# TMS320VC5416

## Fixed-Point Digital Signal Processor

# Data Manual

Literature Number: SPRS095O

March 1999 – Revised January 2005

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of the Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS INSTRUMENTS

## Revision History

This data sheet revision history highlights the technical changes made to the SPRS095N device-specific data sheet to make it an SPRS095O revision.

**Scope:** This document has been reviewed for technical accuracy; the technical content is up-to-date as of the specified release date with the following corrections.

| SECTION | ADDITIONS/CHANGES/DELETIONS |
|---|---|
| Section 5.2 | Changed $I_{OH}$ from -2 to -8 mA and $I_{OL}$ from 2 to 8 mA. Changed Note 2 to read "These output current limits are used for the test conditions on $V_{OL}$ and $V_{OH}$, except where noted otherwise." |
| Chapter 6 | Removed Section 6.2 to prevent duplication of Product Information. |

# Contents

# List of Figures

# List of Tables

# 1 TMS320VC5416 Features

- **Advanced Multibus Architecture With Three Separate 16-Bit Data Memory Buses and One Program Memory Bus**
- **40-Bit Arithmetic Logic Unit (ALU) Including a 40-Bit Barrel Shifter and Two Independent 40-Bit Accumulators**
- **17- × 17-Bit Parallel Multiplier Coupled to a 40-Bit Dedicated Adder for Non-Pipelined Single-Cycle Multiply/Accumulate (MAC) Operation**
- **Compare, Select, and Store Unit (CSSU) for the Add/Compare Selection of the Viterbi Operator**
- **Exponent Encoder to Compute an Exponent Value of a 40-Bit Accumulator Value in a Single Cycle**
- **Two Address Generators With Eight Auxiliary Registers and Two Auxiliary Register Arithmetic Units (ARAUs)**
- **Data Bus With a Bus Holder Feature**
- **Extended Addressing Mode for 8M × 16-Bit Maximum Addressable External Program Space**
- **128K x 16-Bit On-Chip RAM Composed of:**
  - **Eight Blocks of 8K × 16-Bit On-Chip Dual-Access Program/Data RAM**
  - **Eight Blocks of 8K × 16-Bit On-Chip Single-Access Program RAM**
- **16K × 16-Bit On-Chip ROM Configured for Program Memory**
- **Enhanced External Parallel Interface (XIO2)**
- **Single-Instruction-Repeat and Block-Repeat Operations for Program Code**
- **Block-Memory-Move Instructions for Better Program and Data Management**
- **Instructions With a 32-Bit Long Word Operand**
- **Instructions With Two- or Three-Operand Reads**
- **Arithmetic Instructions With Parallel Store and Parallel Load**
- **Conditional Store Instructions**
- **Fast Return From Interrupt**
- **On-Chip Peripherals**
  - **Software-Programmable Wait-State Generator and Programmable Bank-Switching**
  - **On-Chip Programmable Phase-Locked Loop (PLL) Clock Generator With External Clock Source**
  - **One 16-Bit Timer**
  - **Six-Channel Direct Memory Access (DMA) Controller**
  - **Three Multichannel Buffered Serial Ports (McBSPs)**
  - **8/16-Bit Enhanced Parallel Host-Port Interface (HPI8/16)**
- **Power Consumption Control With IDLE1, IDLE2, and IDLE3 Instructions With Power-Down Modes**
- **CLKOUT Off Control to Disable CLKOUT**
- **On-Chip Scan-Based Emulation Logic, IEEE Std 1149.1 (JTAG) Boundary Scan Logic**(1)
- **144-Pin Ball Grid Array (BGA)(GGU Suffix)**
- **144-Pin Low-Profile Quad Flatpack (LQFP)(PGE Suffix)**
- **6.25-ns Single-Cycle Fixed-Point Instruction Execution Time (160 MIPS)**
- **8.33-ns Single-Cycle Fixed-Point Instruction Execution Time (120 MIPS)**
- **3.3-V I/O Supply Voltage (160 and 120 MIPS)**
- **1.6-V Core Supply Voltage (160 MIPS)**
- **1.5-V Core Supply Voltage (120 MIPS)**

(1) IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture

# 2 Introduction

This section describes the main features of the TMS320VC5416, lists the pin assignments, and describes the function of each pin. This data manual also provides a detailed description section, electrical specifications, parameter measurement information, and mechanical data about the available packaging.

**NOTE**

This data manual is designed to be used in conjunction with the *TMS320C54x™ DSP Functional Overview* (literature number SPRU307).

## 2.1 Description

The TMS320VC5416 fixed-point, digital signal processor (DSP) (hereafter referred to as the device unless otherwise specified) is based on an advanced modified Harvard architecture that has one program memory bus and three data memory buses. This processor provides an arithmetic logic unit (ALU) with a high degree of parallelism, application-specific hardware logic, on-chip memory, and additional on-chip peripherals. The basis of the operational flexibility and speed of this DSP is a highly specialized instruction set.

Separate program and data spaces allow simultaneous access to program instructions and data, providing a high degree of parallelism. Two read operations and one write operation can be performed in a single cycle. Instructions with parallel store and application-specific instructions can fully utilize this architecture. In addition, data can be transferred between data and program spaces. Such parallelism supports a powerful set of arithmetic, logic, and bit-manipulation operations that can all be performed in a single machine cycle. The device also includes the control mechanisms to manage interrupts, repeated operations, and function calls.

## 2.2 Pin Assignments

Figure 2-1 illustrates the ball locations for the 144-pin ball grid array (BGA) package and is used in conjunction with Table 2-1 to locate signal names and ball grid numbers. Figure 2-2 provides the pin assignments for the 144-pin low-profile quad flatpack (LQFP) package.

### *2.2.1 Terminal Assignments for the GGU Package*

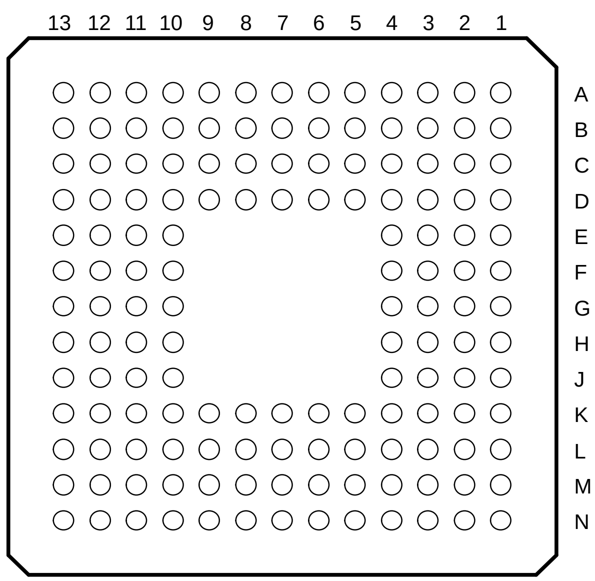



**Figure 2-1. 144-Ball GGU MicroStar BGA™ (Bottom View)**

Table 2-1 lists each signal name and BGA ball number for the 144-pin TMS320VC5416GGU package. Table 2-2 lists each terminal name, terminal function, and operating modes for the TMS320VC5416. In Table 2-1, $DV_{DD}$ is the power supply for the I/O pins while $CV_{DD}$ is the power supply for the core CPU.

**Table 2-1. Terminal Assignments for the TMS320VC5416GGU (144-Pin BGA Package)**

| SIGNAL QUADRANT 1 | BGA BALL # | SIGNAL QUADRANT 2 | BGA BALL # | SIGNAL QUADRANT 3 | BGA BALL # | SIGNAL QUADRANT 4 | BGA BALL # |
|---|---|---|---|---|---|---|---|
| $CV_{SS}$ | A1 | BFSX1 | N13 | $CV_{SS}$ | N1 | A19 | A13 |
| A22 | B1 | BDX1 | M13 | BCLKR1 | N2 | A20 | A12 |
| $CV_{SS}$ | C2 | $DV_{DD}$ | L12 | HCNTL0 | M3 | $CV_{SS}$ | B11 |
| $DV_{DD}$ | C1 | $DV_{SS}$ | L13 | $DV_{SS}$ | N3 | $DV_{DD}$ | A11 |
| A10 | D4 | CLKMD1 | K10 | BCLKR0 | K4 | D6 | D10 |
| HD7 | D3 | CLKMD2 | K11 | BCLKR2 | L4 | D7 | C10 |
| A11 | D2 | CLKMD3 | K12 | BFSR0 | M4 | D8 | B10 |
| A12 | D1 | HPI16 | K13 | BFSR2 | N4 | D9 | A10 |
| A13 | E4 | HD2 | J10 | BDR0 | K5 | D10 | D9 |
| A14 | E3 | TOUT | J11 | HCNTL1 | L5 | D11 | C9 |
| A15 | E2 | EMU0 | J12 | BDR2 | M5 | D12 | B9 |
| $CV_{DD}$ | E1 | EMU1/$\overline{\text{OFF}}$ | J13 | BCLKX0 | N5 | HD4 | A9 |
| $\overline{\text{HAS}}$ | F4 | TDO | H10 | BCLKX2 | K6 | D13 | D8 |
| $DV_{SS}$ | F3 | TDI | H11 | $CV_{SS}$ | L6 | D14 | C8 |
| $CV_{SS}$ | F2 | $\overline{\text{TRST}}$ | H12 | $\overline{\text{HINT}}$ | M6 | D15 | B8 |
| $CV_{DD}$ | F1 | TCK | H13 | $CV_{DD}$ | N6 | HD5 | A8 |
| $\overline{\text{HCS}}$ | G2 | TMS | G12 | BFSX0 | M7 | $CV_{DD}$ | B7 |
| HR/$\overline{\text{W}}$ | G1 | $CV_{SS}$ | G13 | BFSX2 | N7 | $CV_{SS}$ | A7 |
| READY | G3 | $CV_{DD}$ | G11 | HRDY | L7 | $\overline{\text{HDS1}}$ | C7 |
| $\overline{\text{PS}}$ | G4 | HPIENA | G10 | $DV_{DD}$ | K7 | $DV_{SS}$ | D7 |
| $\overline{\text{DS}}$ | H1 | $DV_{SS}$ | F13 | $DV_{SS}$ | N8 | $\overline{\text{HDS2}}$ | A6 |
| $\overline{\text{IS}}$ | H2 | CLKOUT | F12 | HD0 | M8 | $DV_{DD}$ | B6 |
| R/$\overline{\text{W}}$ | H3 | HD3 | F11 | BDX0 | L8 | A0 | C6 |
| $\overline{\text{MSTRB}}$ | H4 | X1 | F10 | BDX2 | K8 | A1 | D6 |
| $\overline{\text{IOSTRB}}$ | J1 | X2/CLKIN | E13 | $\overline{\text{IACK}}$ | N9 | A2 | A5 |
| $\overline{\text{MSC}}$ | J2 | $\overline{\text{RS}}$ | E12 | HBIL | M9 | A3 | B5 |
| XF | J3 | D0 | E11 | $\overline{\text{NMI}}$ | L9 | HD6 | C5 |
| $\overline{\text{HOLDA}}$ | J4 | D1 | E10 | $\overline{\text{INT0}}$ | K9 | A4 | D5 |
| $\overline{\text{IAQ}}$ | K1 | D2 | D13 | $\overline{\text{INT1}}$ | N10 | A5 | A4 |
| $\overline{\text{HOLD}}$ | K2 | D3 | D12 | $\overline{\text{INT2}}$ | M10 | A6 | B4 |
| $\overline{\text{BIO}}$ | K3 | D4 | D11 | $\overline{\text{INT3}}$ | L10 | A7 | C4 |
| MP/$\overline{\text{MC}}$ | L1 | D5 | C13 | $CV_{DD}$ | N11 | A8 | A3 |
| $DV_{DD}$ | L2 | A16 | C12 | HD1 | M11 | A9 | B3 |
| $CV_{SS}$ | L3 | $DV_{SS}$ | C11 | $CV_{SS}$ | L11 | $CV_{DD}$ | C3 |
| BDR1 | M1 | A17 | B13 | BCLKX1 | N12 | A21 | A2 |
| BFSR1 | M2 | A18 | B12 | $DV_{SS}$ | M12 | $DV_{SS}$ | B2 |

### 2.2.2 Pin Assignments for the PGE Package

The TMS320VC5416PGE 144-pin low-profile quad flatpack (LQFP) pin assignments are shown in Figure 2-2. $DV_{DD}$ is the power supply for the I/O pins while $CV_{DD}$ is the power supply for the core CPU. $V_{SS}$ is the ground for both the I/O pins and the core CPU.

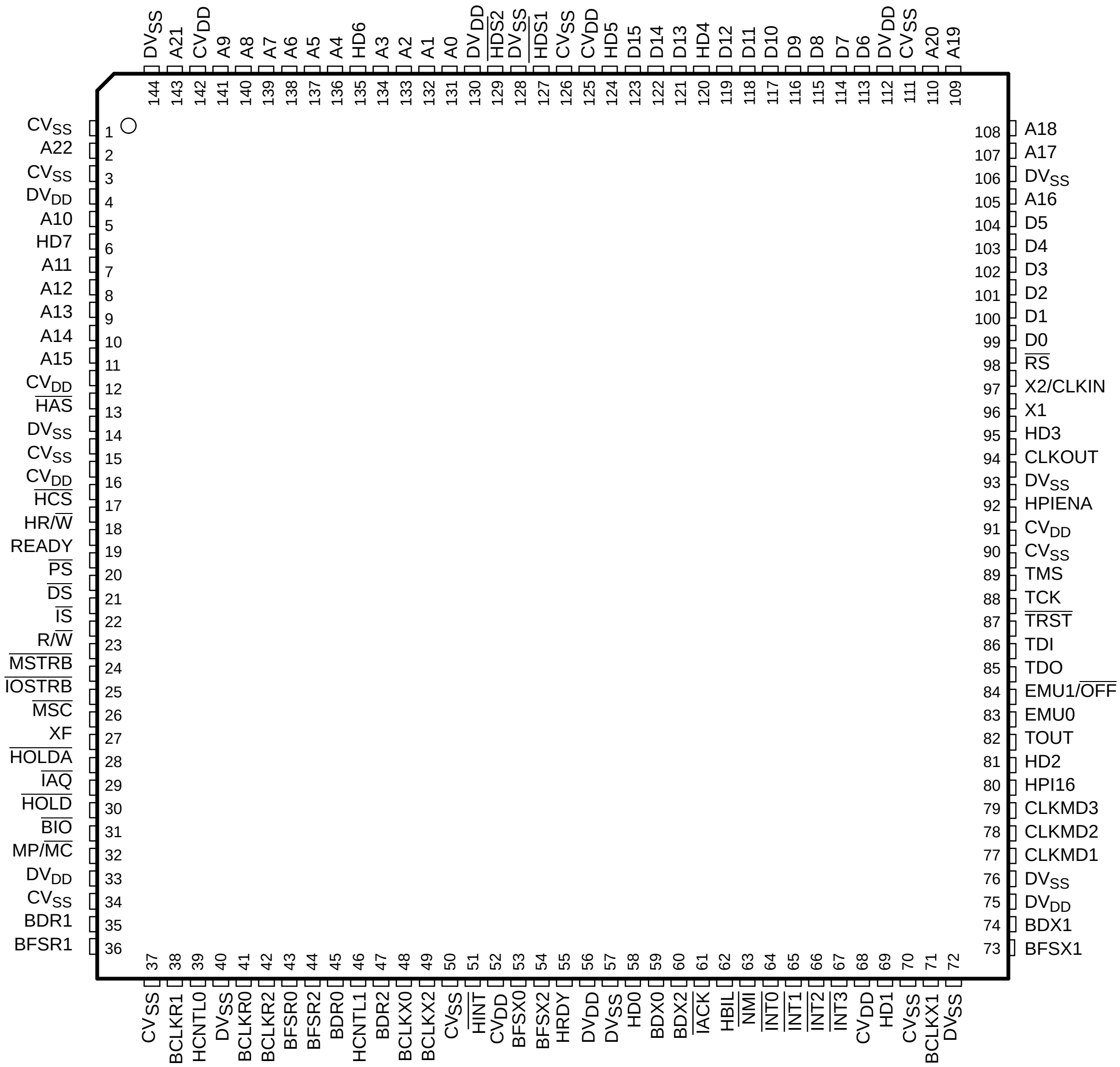



**Figure 2-2. 144-Pin PGE Low-Profile Quad Flatpack (Top View)**

### 2.2.3 Signal Descriptions

Table 2-2 lists each signal, function, and operating mode(s) grouped by function.

**Table 2-2. Signal Descriptions**

| TERMINAL NAME | I/O(1) | DESCRIPTION |
|---|---|---|
| **DATA SIGNALS** | | |
| A22 (MSB), A21, A20, A19, A18, A17, A16, A15, A14, A13, A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0 (LSB) | I/O/Z(2)(3) | Parallel address bus A22 [most significant bit (MSB)] through A0 [least significant bit (LSB)]. The sixteen LSB lines, A0 to A15, are multiplexed to address external memory (program, data) or I/O. The seven MSB lines, A16 to A22, address external program space memory. A22-A0 is placed in the high-impedance state in the hold mode. A22-A0 also goes into the high-impedance state when $\overline{OFF}$ is low.A17-A0 are inputs in HPI16 mode. These pins can be used to address internal memory via the host-port interface (HPI) when the HPI16 pin is high. These pins also have Schmitt trigger inputs.The address bus has a bus holder feature that eliminates passive components and the power dissipation associated with them. The bus holder keeps the address bus at the previous logic level when the bus goes into a high-impedance state. |
| D15 (MSB), D14, D13, D12, D11, D10, D9, D8, D7, D6, D5, D4, D3, D2, D1, D0 (LSB) | I/O/Z(2)(3) | Parallel data bus D15 (MSB) through D0 (LSB). D15-D0 is multiplexed to transfer data between the core CPU and external data/program memory or I/O devices or HPI in HPI16 mode (when HPI16 pin is high). D15-D0 is placed in the high-impedance state when not outputting data or when $\overline{RS}$ or $\overline{HOLD}$ is asserted. D15-D0 also goes into the high-impedance state when $\overline{OFF}$ is low. These pins also have Schmitt trigger inputs.The data bus has a bus holder feature that eliminates passive components and the power dissipation associated with them. The bus holder keeps the data bus at the previous logic level when the bus goes into the high-impedance state. The bus holders on the data bus can be enabled/disabled under software control. |
| **INITIALIZATION, INTERRUPT AND RESET OPERATIONS** | | |
| $\overline{IACK}$ | O/Z | Interrupt acknowledge signal. $\overline{IACK}$ indicates receipt of an interrupt and that the program counter is fetching the interrupt vector location designated by A15-A0. $\overline{IACK}$ also goes into the high-impedance state when $\overline{OFF}$ is low. |
| $\overline{INT0}$ (2)<br>$\overline{INT1}$ (2)<br>$\overline{INT2}$ (2)<br>$\overline{INT3}$ (2) | I | External user interrupt inputs. $\overline{INT0}$-$\overline{INT3}$ are maskable and are prioritized by the interrupt mask register (IMR) and the interrupt mode bit. $\overline{INT0}$ -$\overline{INT3}$ can be polled and reset by way of the interrupt flag register (IFR). |
| $\overline{NMI}$ (2) | I | Nonmaskable interrupt. $\overline{NMI}$ is an external interrupt that cannot be masked by way of the INTM or the IMR. When $\overline{NMI}$ is activated, the processor traps to the appropriate vector location. |
| $\overline{RS}$ (2) | I | Reset. $\overline{RS}$ causes the digital signal processor (DSP) to terminate execution and forces the program counter to 0FF80h. When $\overline{RS}$ is brought to a high level, execution begins at location 0FF80h of program memory. $\overline{RS}$ affects various registers and status bits. |
| MP/$\overline{MC}$ | I | Microprocessor/microcomputer mode select. If active low at reset, microcomputer mode is selected, and the internal program ROM is mapped into the upper 16K words of program memory space. If the pin is driven high during reset, microprocessor mode is selected, and the on-chip ROM is removed from program space. This pin is only sampled at reset, and the MP/$\overline{MC}$ bit of the processor mode status (PMST) register can override the mode that is selected at reset. |
| **MULTIPROCESSING SIGNALS** | | |
| $\overline{BIO}$ (2) | I | Branch control. A branch can be conditionally executed when $\overline{BIO}$ is active. If low, the processor executes the conditional instruction. The $\overline{BIO}$ condition is sampled during the decode phase of the pipeline for the XC instruction, and all other instructions sample $\overline{BIO}$ during the read phase of the pipeline. |
| XF | O/Z | External flag output (latched software-programmable signal). XF is set high by the SSBX XF instruction, set low by RSBX XF instruction or by loading ST1. XF is used for signaling other processors in multiprocessor configurations or used as a general-purpose output pin. XF goes into the high-impedance state when $\overline{OFF}$ is low, and is set high at reset. |
| **MEMORY CONTROL SIGNALS** | | |
| $\overline{DS}$<br>$\overline{PS}$<br>$\overline{IS}$ | O/Z | Data, program, and I/O space select signals. $\overline{DS}$, $\overline{PS}$, and $\overline{IS}$ are always high unless driven low for communicating to a particular external space. Active period corresponds to valid address information. $\overline{DS}$, $\overline{PS}$, and $\overline{IS}$ are placed into the high-impedance state in the hold mode; these signals also go into the high-impedance state when $\overline{OFF}$ is low. |
| $\overline{MSTRB}$ | O/Z | Memory strobe signal. $\overline{MSTRB}$ is always high unless low-level asserted to indicate an external bus access to data or program memory. $\overline{MSTRB}$ is placed in the high-impedance state in the hold mode; it also goes into the high-impedance state when $\overline{OFF}$ is low. |

(1) I = Input, O = Output, Z = High-impedance, S = Supply
(2) These pins have Schmitt trigger inputs.
(3) This pin has an internal bus holder controlled by way of the BSCR register.

### Table 2-2. Signal Descriptions (continued)

| TERMINAL NAME | I/O(1) | DESCRIPTION |
|---|---|---|
| READY | I | Data ready. READY indicates that an external device is prepared for a bus transaction to be completed. If the device is not ready (READY is low), the processor waits one cycle and checks READY again. Note that the processor performs ready detection if at least two software wait states are programmed. The READY signal is not sampled until the completion of the software wait states. |
| R/$\overline{\text{W}}$ | O/Z | Read/write signal. R/$\overline{\text{W}}$ indicates transfer direction during communication to an external device. R/$\overline{\text{W}}$ is normally in the read mode (high), unless it is asserted low when the DSP performs a write operation. R/$\overline{\text{W}}$ is placed in the high-impedance state in the hold mode; and it also goes into the high-impedance state when $\overline{\text{OFF}}$ is low. |
| $\overline{\text{IOSTRB}}$ | O/Z | I/O strobe signal. $\overline{\text{IOSTRB}}$ is always high unless low-level asserted to indicate an external bus access to an I/O device. $\overline{\text{IOSTRB}}$ is placed in the high-impedance state in the hold mode; it also goes into the high-impedance state when $\overline{\text{OFF}}$ is low. |
| $\overline{\text{HOLD}}$ | I | Hold input. $\overline{\text{HOLD}}$ is asserted to request control of the address, data, and control lines. When acknowledged by the device, these lines go into the high-impedance state. |
| $\overline{\text{HOLDA}}$ | O/Z | Hold acknowledge. $\overline{\text{HOLDA}}$ indicates to the external circuitry that the processor is in a hold state and that the address, data, and control lines are in the high-impedance state, allowing them to be available to the external circuitry. $\overline{\text{HOLDA}}$ also goes into the high-impedance state when $\overline{\text{OFF}}$ is low. Figure 2-2 This pin is driven high during reset. |
| $\overline{\text{MSC}}$ | O/Z | Microstate complete. $\overline{\text{MSC}}$ indicates completion of all software wait states. When two or more software wait states are enabled, the $\overline{\text{MSC}}$ pin goes active at the beginning of the first software wait state and goes inactive high at the beginning of the last software wait state. If connected to the READY input, $\overline{\text{MSC}}$ forces one external wait state after the last internal wait state is completed. $\overline{\text{MSC}}$ also goes into the high-impedance state when $\overline{\text{OFF}}$ is low. |
| $\overline{\text{IAQ}}$ | O/Z | Instruction acquisition signal. $\overline{\text{IAQ}}$ is asserted (active low) when there is an instruction address on the address bus and goes into the high-impedance state when $\overline{\text{OFF}}$ is low. |
| **TIMER SIGNALS** | | |
| CLKOUT | O/Z | Clock output signal. CLKOUT can represent the machine-cycle rate of the CPU divided by 1, 2, 3, or 4 as configured in the bank-switching control register (BSCR). Following reset, CLKOUT represents the machine-cycle rate divided by 4. |
| CLKMD1(2)<br>CLKMD2(2)<br>CLKMD3(2) | I | Clock mode select signals. CLKMD1-CLKMD3 allow the selection and configuration of different clock modes such as crystal, external clock, and PLL mode. The external CLKMD1-CLKMD3 pins are sampled to determine the desired clock generation mode while $\overline{\text{RS}}$ is low. Following reset, the clock generation mode can be reconfigured by writing to the internal clock mode register in software. |
| X2/CLKIN(2) | I | Clock/oscillator input. If the internal oscillator is not being used, X2/CLKIN functions as the clock input. (This is revision-dependent, see Section Section 3.10 for additional information.) |
| X1 | O | Output pin from the internal oscillator for the crystal. If the internal oscillator is not used, X1 should be left unconnected. X1 does not go into the high-impedance state when $\overline{\text{OFF}}$ is low. (This is revision-dependent, see Section Section 3.10 for additional information.) |
| TOUT | O/Z | Timer output. TOUT signals a pulse when the on-chip timer counts down past zero. The pulse is one CLKOUT cycle wide. TOUT also goes into the high-impedance state when $\overline{\text{OFF}}$ is low. |
| **MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP #0), MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP #1),AND MULTICHANNEL BUFFERED SERIAL PORT 2 (McBSP #2) SIGNALS** | | |
| BCLKR0(2)<br>BCLKR1(2)<br>BCLKR2(2) | I/O/Z | Receive clock input. BCLKR can be configured as an input or an output; it is configured as an input following reset. BCLKR serves as the serial shift clock for the buffered serial port receiver. |
| BDR0, BDR1, BDR2 | I | Serial data receive input |
| BFSR0, BFSR1, BFSR2 | I/O/Z | Frame synchronization pulse for receive input. BFSR can be configured as an input or an output; it is configured as an input following reset. The BFSR pulse initiates the receive data process over BDR. |
| BCLKX0(2)<br>BCLKX1(2)<br>BCLKX2(2) | I/O/Z | Transmit clock. BCLKX serves as the serial shift clock for the McBSP transmitter. BCLKX can be configured as an input or an output, and is configured as an input following reset. BCLKX enters the high-impedance state when $\overline{\text{OFF}}$ goes low. |
| BDX0, BDX1, BDX2 | O/Z | Serial data transmit output. BDX is placed in the high-impedance state when not transmitting, when $\overline{\text{RS}}$ is asserted, or when $\overline{\text{OFF}}$ is low. |
| BFSX0, BFSX1, BFSX2 | I/O/Z | Frame synchronization pulse for transmit input/output. The BFSX pulse initiates the data transmit process over BDX. BFSX can be configured as an input or an output, and is configured as an input following reset. BFSX goes into the high-impedance state when $\overline{\text{OFF}}$ is low. |

### Table 2-2. Signal Descriptions (continued)

| TERMINAL NAME | I/O[1] | DESCRIPTION |
|---|---|---|
| **HOST-PORT INTERFACE SIGNALS** | | |
| HD0-HD7[2][3] | I/O/Z | Parallel bidirectional data bus. The HPI data bus is used by a host device bus to exchange information with the HPI registers. These pins can also be used as general-purpose I/O pins. HD0-HD7 is placed in the high-impedance state when not outputting data or when $\overline{\text{OFF}}$ is low. The HPI data bus includes bus holders to reduce the static power dissipation caused by floating, unused pins. When the HPI data bus is not being driven by the device, the bus holders keep the pins at the previous logic level. The HPI data bus holders are disabled at reset and can be enabled/disabled via the HBH bit of the BSCR. These pins also have Schmitt trigger inputs. |
| HCNTL0[4]<br>HCNTL1[4] | I | Control inputs. HCNTL0 and HCNTL1 select a host access to one of the three HPI registers. The control inputs have internal pullups that are only enabled when HPIENA = 0. These pins are not used when HPI16 = 1. |
| HBIL[4] | I | Byte identification. HBIL identifies the first or second byte of transfer. The HPIL input has an internal pullup resistor that is only enabled when HPIENA = 0. This pin is not used when HPI16 = 1. |
| $\overline{\text{HCS}}$ [2] [4] | I | Chip select. $\overline{\text{HCS}}$ is the select input for the HPI and must be driven low during accesses. The chip select input has an internal pullup resistor that is only enabled when HPIENA = 0. |
| $\overline{\text{HDS1}}$ [2] [4]<br>$\overline{\text{HDS2}}$ [2] [4] | I | Data strobe. $\overline{\text{HDS1}}$ and $\overline{\text{HDS2}}$ are driven by the host read and write strobes to control the transfer. The strobe inputs have internal pullup resistors that are only enabled when HPIENA = 0. |
| $\overline{\text{HAS}}$ [2] [4] | I | Address strobe. Host with multiplexed address and data pins requires $\overline{\text{HAS}}$ to latch the address in the HPIA register. $\overline{\text{HAS}}$ input has an internal pullup resistor that is only enabled when HPIENA = 0. |
| HR/$\overline{\text{W}}$[4] | I | Read/write. HR/$\overline{\text{W}}$ controls the direction of the HPI transfer. HR/$\overline{\text{W}}$ has an internal pullup resistor that is only enabled when HPIENA = 0. |
| HRDY | O/Z | Ready output. HRDY goes into the high-impedance state when $\overline{\text{OFF}}$ is low. The ready output informs the host when the HPI is ready for the next transfer. This pin is driven high during reset. |
| $\overline{\text{HINT}}$ | O/Z | Interrupt output. This output is used to interrupt the host. When the DSP is in reset, $\overline{\text{HINT}}$ is driven high. $\overline{\text{HINT}}$ goes into the high-impedance state when $\overline{\text{OFF}}$ is low. This pin is not used when HPI16 = 1. |
| HPIENA[5] | I | HPI module select. HPIENA must be tied to $DV_{DD}$ to have HPI selected. If HPIENA is left open or connected to ground, the HPI module is not selected, internal pullup for the HPI input pins are enabled, and the HPI data bus has holders set. HPIENA is provided with an internal pulldown resistor that is always active. HPIENA is sampled when $\overline{\text{RS}}$ goes high and is ignored until $\overline{\text{RS}}$ goes low again. This pin should never be changed while reset is high. |
| HPI16[5] | I | HPI16 mode selection. This pin must be tied to DVDD to enable HPI16 mode. The pin has an internal pulldown resistor which is always active. If HPI16 is left open or driven low, the HPI16 mode is disabled. |
| **SUPPLY PINS** | | |
| $CV_{SS}$ | S | Ground. Dedicated ground for the core CPU |
| $CV_{DD}$ | S | $+V_{DD}$. Dedicated power supply for the core CPU |
| $DV_{SS}$ | S | Ground. Dedicated ground for I/O pins |
| $DV_{DD}$ | S | $+V_{DD}$. Dedicated power supply for I/O pins |
| **TEST PINS** | | |
| TCK[2][4] | I | IEEE standard 1149.1 test clock. TCK is normally a free-running clock signal with a 50% duty cycle. The changes on test access port (TAP) of input signals TMS and TDI are clocked into the TAP controller, instruction register, or selected test data register on the rising edge of TCK. Changes at the TAP output signal (TDO) occur on the falling edge of TCK. |
| TDI[4] | I | IEEE standard 1149.1 test data input. Pin with internal pullup device. TDI is clocked into the selected register (instruction or data) on a rising edge of TCK. |
| TDO | O/Z | IEEE standard 1149.1 test data output. The contents of the selected register (instruction or data) are shifted out of TDO on the falling edge of TCK. TDO is in the high-impedance state except when the scanning of data is in progress. TDO also goes into the high-impedance state when $\overline{\text{OFF}}$ is low. |
| TMS[4] | I | IEEE standard 1149.1 test mode select. Pin with internal pullup device. This serial control input is clocked into the TAP controller on the rising edge of TCK. |
| $\overline{\text{TRST}}$ [5] | I | IEEE standard 1149.1 test reset. $\overline{\text{TRST}}$, when high, gives the IEEE standard 1149.1 scan system control of the operations of the device. If $\overline{\text{TRST}}$ is not connected or driven low, the device operates in its functional mode, and the IEEE standard 1149.1 signals are ignored. Pin with internal pulldown device. |
| EMU0[6] | I/O/Z | Emulator 0 pin. When $\overline{\text{TRST}}$ is driven low, EMU0 must be high for activation of the $\overline{\text{OFF}}$ condition. When $\overline{\text{TRST}}$ is driven high, EMU0 is used as an interrupt to or from the emulator system and is defined as input/output by way of the IEEE standard 1149.1 scan system. |

(4) This pin has an internal pullup resistor.
(5) This pin has an internal pulldown resistor.
(6) This pin must be pulled up with a 4.7-kΩ resistor to ensure the device is operable in functional mode or emulation mode.

**Table 2-2. Signal Descriptions (continued)**

| TERMINAL NAME | I/O[1] | DESCRIPTION |
|---|---|---|
| EMU1/$\overline{\text{OFF}}$[6] | I/O/Z | Emulator 1 pin/disable all outputs. When $\overline{\text{TRST}}$ is driven high, EMU1/$\overline{\text{OFF}}$ is used as an interrupt to or from the emulator system and is defined as input/output by way of IEEE standard 1149.1 scan system. When $\overline{\text{TRST}}$ is driven low, EMU1/$\overline{\text{OFF}}$ is configured as $\overline{\text{OFF}}$. The EMU1/$\overline{\text{OFF}}$ signal, when active low, puts all output drivers into the high-impedance state. Note that $\overline{\text{OFF}}$ is used exclusively for testing and emulation purposes (not for multiprocessing applications). Therefore, for the $\overline{\text{OFF}}$ condition, the following apply:<br>$\overline{\text{TRST}}$ = low,<br>EMU0 = high<br>EMU1/$\overline{\text{OFF}}$ = low |

# 3 Functional Overview

The following functional overview is based on the block diagram in Figure 3-1.

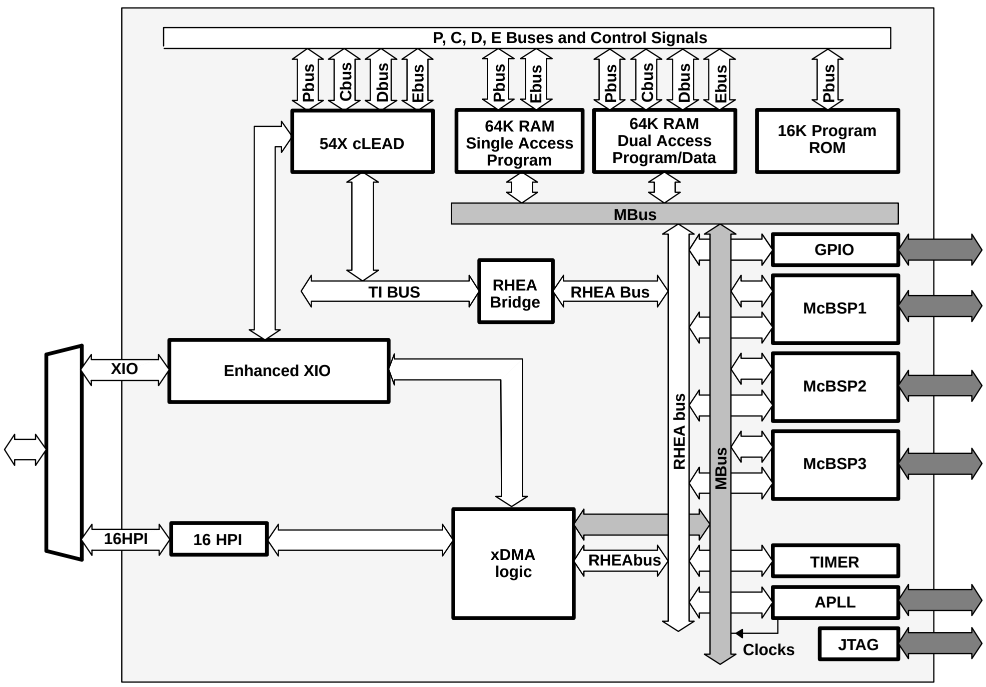



**Figure 3-1. TMS320VC5416 Functional Block Diagram**

## 3.1 Memory

The device provides both on-chip ROM and RAM memories to aid in system performance and integration.

### *3.1.1 Data Memory*

The data memory space addresses up to 64K of 16-bit words. The device automatically accesses the on-chip RAM when addressing within its bounds. When an address is generated outside the RAM bounds, the device automatically generates an external access.

The advantages of operating from on-chip memory are as follows:

- Higher performance because no wait states are required
- Higher performance because of better flow within the pipeline of the central arithmetic logic unit (CALU)
- Lower cost than external memory

- Lower power than external memory

The advantage of operating from off-chip memory is the ability to access a larger address space.

### *3.1.2 Program Memory*

Software can configure their memory cells to reside inside or outside of the program address map. When the cells are mapped into program space, the device automatically accesses them when their addresses are within bounds. When the program-address generation (PAGEN) logic generates an address outside its bounds, the device automatically generates an external access. The advantages of operating from on-chip memory are as follows:

- Higher performance because no wait states are required
- Lower cost than external memory
- Lower power than external memory

The advantage of operating from off-chip memory is the ability to access a larger address space.

### *3.1.3 Extended Program Memory*

The device uses a paged extended memory scheme in program space to allow access of up to 8192K of program memory. In order to implement this scheme, the device includes several features which are also present on C548/549/5410:

- Twenty-three address lines, instead of sixteen
- An extra memory-mapped register, the XPC
- Six extra instructions for addressing extended program space

Program memory in the device is organized into 128 pages that are each 64K in length.

The value of the XPC register defines the page selection. This register is memory-mapped into data space to address 001Eh. At a hardware reset, the XPC is initialized to 0.

## 3.2 On-Chip ROM With Bootloader

The device features a 16K-word × 16-bit on-chip maskable ROM that can only be mapped into program memory space.

Customers can arrange to have the ROM of the device programmed with contents unique to any particular application.

A bootloader is available in the standard on-chip ROM. This bootloader can be used to automatically transfer user code from an external source to anywhere in the program memory at power up. If MP/$\overline{\text{MC}}$ of the device is sampled low during a hardware reset, execution begins at location FF80h of the on-chip ROM. This location contains a branch instruction to the start of the bootloader program.

The standard devices provide different ways to download the code to accommodate various system requirements:

- Parallel from 8-bit or 16-bit-wide EPROM
- Parallel from I/O space, 8-bit or 16-bit mode
- Serial boot from serial ports, 8-bit or 16-bit mode
- Host-port interface boot
- Warm boot

The standard on-chip ROM layout is shown in Table 3-1.

**Table 3-1. Standard On-Chip ROM Layout**

| ADDRESS RANGE | DESCRIPTION |
|---|---|
| C000h-D4FFh | ROM tables for the GSM EFR speech codec |
| D500h-F7FFh | Reserved |
| F800h-FBFFh | Bootloader |
| FC00h-FCFFh | µ-Law expansion table |
| FD00h-FDFFh | A-Law expansion table |
| FE00h-FEFFh | Sine look-up table |
| FF00h-FF7Fh | Reserved(1) |
| FF80h-FFFFh | Interrupt vector table |

(1) In the ROM, 128 words are reserved for factory device-testing purposes. Application code to be implemented in on-chip ROM must reserve these 128 words at addresses FF00h-FF7Fh in program space.

## 3.3 On-Chip RAM

The device contains 64K-word × 16-bit of on-chip dual-access RAM (DARAM) and 64K-word × 16-bit of on-chip single-access RAM (SARAM).

The DARAM is composed of eight blocks of 8K words each. Each block in the DARAM can support two reads in one cycle, or a read and a write in one cycle. Four blocks of DARAM are located in the address range 0080h-7FFFh in data space, and can be mapped into program/data space by setting the OVLY bit to one. The other four blocks of DARAM are located in the address range 18000h-1FFFFh in program space. The DARAM located in the address range 18000h-1FFFFh in program space can be mapped into data space by setting the DROM bit to one.

The SARAM is composed of eight blocks of 8K words each. Each of these eight blocks is a single-access memory. For example, an instruction word can be fetched from one SARAM block in the same cycle as a data word is written to another SARAM block. The SARAM is located in the address range 28000h-2FFFFh, and 38000h-3FFFFh in program space.

## 3.4 On-Chip Memory Security

The device has a maskable option to protect the contents of on-chip memories.

When the *RAM/ROM* security option is selected, the following restrictions apply:

- Only the on-chip ROM originating instructions can read the contents of the on-chip ROM; on-chip RAM and external RAM originating instruction can not read data from ROM: instead 0FFFFh is read. Code can still branch to ROM from on-chip RAM or external program memory.
- The contents of on-chip RAM can be read by all instructions, even by instructions fetched from external memory. To protect the internal RAM, the user must never branch to external memory.
- The security feature completely disables the scan-based emulation capability of the 54x to prevent the use of a debugger utility. This only affects emulation and does not prevent the use of the JTAG boundary scan test capability.
- The device is internally forced into microcomputer mode at reset (MP/$\overline{\text{MC}}$ bit forced to zero), preventing the ROM from being disabled by the external MP/$\overline{\text{MC}}$ pin. The status of the MP/$\overline{\text{MC}}$ bit in the PMST register can be changed after reset by the user application.
- HPI writes have no restriction, but HPI reads are restricted to the 4000h - 5FFFh address range.

If the *ROM*-only security option is selected the following restrictions apply:

- Only the on-chip ROM originating instructions can read the contents of the on-chip ROM; on-chip RAM and external RAM originating instruction cannot read data from ROM: instead 0FFFFh is read. Code can still branch to ROM from on-chip RAM or external program memory.
- The contents of on-chip RAM can be read by all instructions, even by instructions fetched from external memory. To protect the internal RAM the user must never branch to external memory.
- The security feature completely disables the scan-based emulation capability of the 54x to prevent the use of a debugger utility. This only affects emulation and does not prevent the use of the JTAG boundary scan test capability.
- The device can be started in either microcomputer mode or microprocessor mode at reset (depends on the MP/$\overline{\text{MC}}$ pin).
- HPI read and writes have no restriction.

## 3.5 Memory Map

The program and data memory map is shown in Figure 3-2. Address ranges for on-chip DARAM in data memory are:

- DARAM0: 0080h-1FFFh
- DARAM1: 2000h-3FFFh
- DARAM2: 4000h-5FFFh
- DARAM3: 6000h-7FFFh
- DARAM4: 8000h-9FFFh
- DARAM5: A000h-BFFFh
- DARAM6: C000h-DFFFh
- DARAM7: E000h-FFFFh

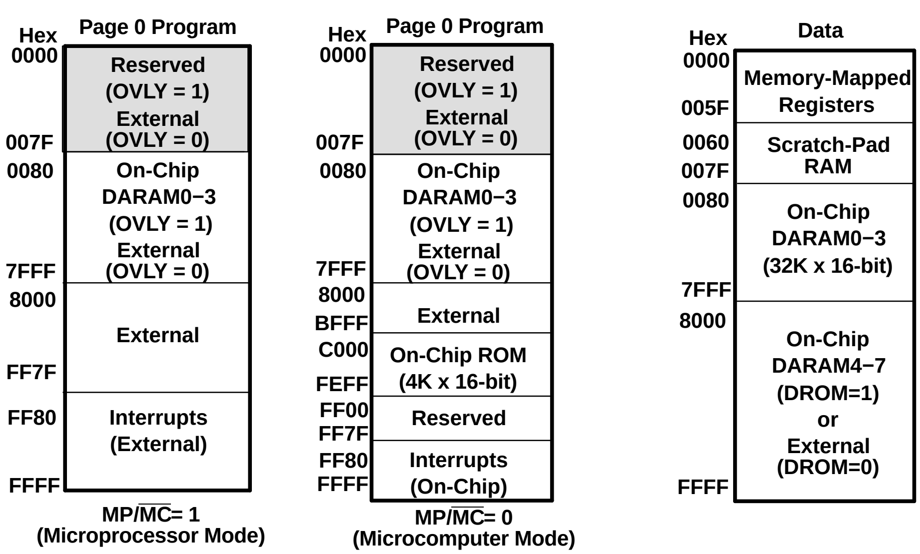



**Figure 3-2. Program and Data Memory Map**

The extended program memory map is shown in Figure 3-3. Address ranges for on-chip DARAM in data memory are:

- DARAM4: 018000h-019FFFh
- DARAM5: 01A000h-01BFFFh
- DARAM6: 01C000h-01DFFFh
- DARAM7: 01E000h-01FFFFh

Address ranges for on-chip SARAM in program memory are:

- SARAM0: 028000h-029FFFh
- SARAM1: 02A000h-02BFFFh
- SARAM2: 02C000h-02DFFFh
- SARAM3: 02E000h-02FFFFh
- SARAM4: 038000h-039FFFh
- SARAM5: 03A000h-03BFFFh
- SARAM6: 03C000h-03DFFFh
- SARAM7: 03E000h-03FFFFh

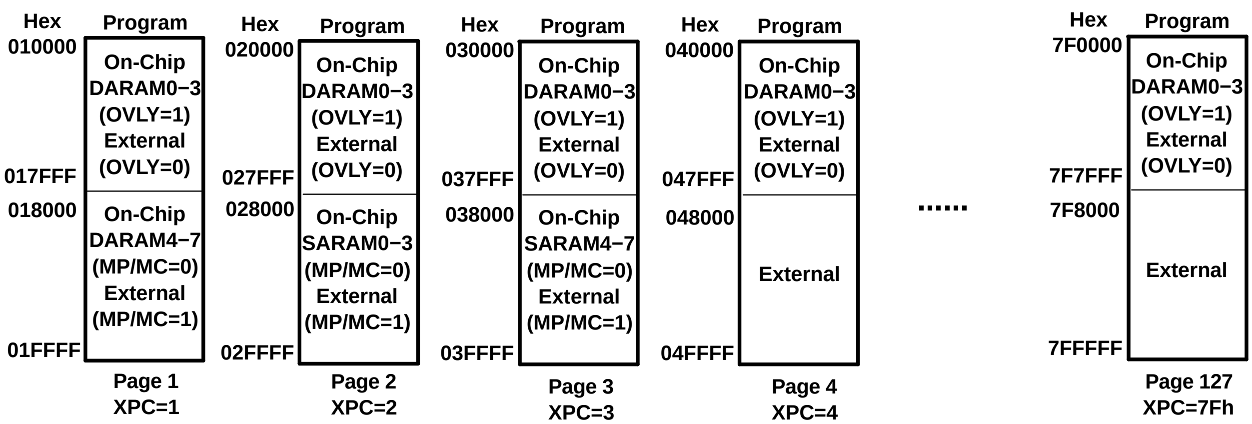



**Figure 3-3. Extended Program Memory Map**

### *3.5.1 Relocatable Interrupt Vector Table*

The reset, interrupt, and trap vectors are addressed in program space. These vectors are soft — meaning that the processor, when taking the trap, loads the program counter (PC) with the trap address and executes the code at the vector location. Four words, either two 1-word instructions or one 2-word instruction, are reserved at each vector location to accommodate a delayed branch instruction which allows branching to the appropriate interrupt service routine without the overhead.

At device reset, the reset, interrupt, and trap vectors are mapped to address FF80h in program space. However, these vectors can be remapped to the beginning of any 128-word page in program space after device reset. This is done by loading the interrupt vector pointer (IPTR) bits in the PMST register with the appropriate 128-word page boundary address. After loading IPTR, any user interrupt or trap vector is mapped to the new 128-word page.

**NOTE**

The hardware reset ($\overline{RS}$) vector cannot be remapped because the hardware reset loads the IPTR with 1s. Therefore, the reset vector is always fetched at location FF80h in program space.

| 15 – 8 |
|---|
| IPTR |
| R/W-1FF |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| IPTR | MP/$\overline{MC}$ | OVLY | AVIS | DROM | CLKOFF | SMUL | SST |
| | MP/$\overline{MC}$ pin | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-4. Process Mode Status Register**

**Table 3-2. Processor Mode Status (PMST) Register Bit Fields**

| BIT NO. | BIT NAME | RESET VALUE | FUNCTION |
|---|---|---|---|
| 15-7 | IPTR | 1FFh | Interrupt vector pointer. The 9-bit IPTR field points to the 128-word program page where the interrupt vectors reside. The interrupt vectors can be remapped to RAM for boot-loaded operations. At reset, these bits are all set to 1; the reset vector always resides at address FF80h in program memory space. The RESET instruction does not affect this field. |
| 6 | MP/$\overline{MC}$ | MP/$\overline{MC}$ pin | Microprocessor/microcomputer mode. MP/$\overline{MC}$ enables/disables the on-chip ROM to be addressable in program memory space.<br>• MP/$\overline{MC}$ = 0: The on-chip ROM is enabled and addressable.<br>• MP/$\overline{MC}$ = 1: The on-chip ROM is not available.<br>MP/$\overline{MC}$ is set to the value corresponding to the logic level on the MP/$\overline{MC}$ pin when sampled at reset. This pin is not sampled again until the next reset. The RESET instruction does not affect this bit. This bit can also be set or cleared by software. |
| 5 | OVLY | 0 | RAM overlay. OVLY enables on-chip dual-access data RAM blocks to be mapped into program space. The values for the OVLY bit are:<br>• OVLY = 0: The on-chip RAM is addressable in data space but not in program space.<br>• OVLY = 1: The on-chip RAM is mapped into program space and data space. Data page 0 (addresses 0h to 7Fh), however, is not mapped into program space. |
| 4 | AVIS | 0 | Address visibility mode. AVIS enables/disables the internal program address to be visible at the address pins.<br>• AVIS = 0: The external address lines do not change with the internal program address. Control and data lines are not affected and the address bus is driven with the last address on the bus.<br>• AVIS = 1: This mode allows the internal program address to appear at the pins of the device so that the internal program address can be traced. Also, it allows the interrupt vector to be decoded in conjunction with $\overline{IACK}$ when the interrupt vectors reside on on-chip memory. |
| 3 | DROM | 0 | DROM enables on-chip DARAM4-7 to be mapped into data space. The DROM bit values are:<br>• DROM = 0: The on-chip DARAM4-7 is not mapped into data space.<br>• DROM = 1: The on-chip DARAM4-7 is mapped into data space. |
| 2 | CLKOFF | 0 | CLOCKOUT off. When the CLKOFF bit is 1, the output of CLKOUT is disabled and remains at a high level. |
| 1 | SMUL | N/A | Saturation on multiplication. When SMUL = 1, saturation of a multiplication result occurs before performing the accumulation in a MAC of MAS instruction. The SMUL bit applies only when OVM = 1 and FRCT = 1. |
| 0 | SST | N/A | Saturation on store. When SST = 1, saturation of the data from the accumulator is enabled before storing in memory. The saturation is performed after the shift operation. |

## 3.6 On-Chip Peripherals

The device has the following peripherals:

- Software-programmable wait-state generator
- Programmable bank-switching
- A host-port interface (HPI8/16)
- Three multichannel buffered serial ports (McBSPs)
- A hardware timer
- A clock generator with a multiple phase-locked loop (PLL)
- Enhanced external parallel interface (XIO2)
- A DMA controller (DMA)

### *3.6.1 Software-Programmable Wait-State Generator*

The software wait-state generator of the device can extend external bus cycles by up to fourteen machine cycles. Devices that require more than fourteen wait states can be interfaced using the hardware READY line. When all external accesses are configured for zero wait states, the internal clocks to the wait-state generator are automatically disabled. Disabling the wait-state generator clocks reduces the power consumption.

The software wait-state register (SWWSR) controls the operation of the wait-state generator. The 14 LSBs of the SWWSR specify the number of wait states (0 to 7) to be inserted for external memory accesses to five separate address ranges. This allows a different number of wait states for each of the five address ranges. Additionally, the software wait-state multiplier (SWSM) bit of the software wait-state control register (SWCR) defines a multiplication factor of 1 or 2 for the number of wait states. At reset, the wait-state generator is initialized to provide seven wait states on all external memory accesses. The SWWSR bit fields are shown in Figure 3-5 and described in Table 3-3.

| 15 | 14–12 | 11–9 | 8 |
|---|---|---|---|
| XPA | I/O | DATA | DATA |
| R/W-0 | R/W-111 | R/W-111 | |

| 7–6 | 5–3 | 2–0 |
|---|---|---|
| DATA | PROGRAM | PROGRAM |
| R/W-111 | R/W-111 | R/W-111 |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-5. Software Wait-State Register (SWWSR) [Memory-Mapped Register (MMR) Address 0028h]**

### Table 3-3. Software Wait-State Register (SWWSR) Bit Fields

| BIT NO. | BIT NAME | RESET VALUE | FUNCTION |
|---|---|---|---|
| 15 | XPA | 0 | Extended program address control bit. XPA is used in conjunction with the program space fields (bits 0 through 5) to select the address range for program space wait states. |
| 14-12 | I/O | 111 | I/O space. The field value (0-7) corresponds to the base number of wait states for I/O space accesses within addresses 0000-FFFFh. The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states. |
| 11-9 | Data | 111 | Upper data space. The field value (0-7) corresponds to the base number of wait states for external data space accesses within addresses 8000-FFFFh. The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states. |
| 8-6 | Data | 111 | Lower data space. The field value (0-7) corresponds to the base number of wait states for external data space accesses within addresses 0000-7FFFh. The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states. |
| 5-3 | Program | 111 | Upper program space. The field value (0-7) corresponds to the base number of wait states for external program space accesses within the following addresses:<br>• XPA = 0: xx8000 - xxFFFFh<br>• XPA = 1: 400000h - 7FFFFFh<br>The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states. |
| 2-0 | Program | 111 | Program space. The field value (0-7) corresponds to the base number of wait states for external program space accesses within the following addresses:<br>• XPA = 0: xx0000 - xx7FFFh<br>• XPA = 1: 000000 - 3FFFFFh.<br>The SWSM bit of the SWCR defines a multiplication factor of 1 or 2 for the base number of wait states. |

The software wait-state multiplier bit of the software wait-state control register (SWCR) is used to extend the base number of wait states selected by the SWWSR. The SWCR bit fields are shown in Figure 3-6 and described in Table 3-4.

| 15 – 8 |
|---|
| Reserved |
| R/W-0 |

| 7 – 1 | 0 |
|---|---|
| Reserved | SWSM |
| R/W-0 | R/W-0 |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-6. Software Wait-State Register (SWWSR) [Memory-Mapped Register (MMR) Address 0028h]**

### Table 3-4. Software Wait-State Control Register (SWCR) Bit Fields

| PIN NO. | PIN NAME | RESET VALUE | FUNCTION |
|---|---|---|---|
| 15-1 | Reserved | 0 | These bits are reserved and are unaffected by writes. |
| 0 | SWSM | 0 | Software wait-state multiplier. Used to multiply the number of wait states defined in the SWWSR by a factor of 1 or 2.<br>• SWSM = 0: wait-state base values are unchanged (multiplied by 1)<br>• SWSM = 1: wait-state base values are multiplied by 2 for a maximum of 14 wait states |

### 3.6.2 Programmable Bank-Switching

Programmable bank-switching logic allows the device to switch between external memory banks without requiring external wait states for memories that need additional time to turn off. The bank-switching logic automatically inserts one cycle when accesses cross a 32K-word memory-bank boundary inside program or data space.

Bank-switching is defined by the bank-switching control register (BSCR), which is memory-mapped ataddress 0029h. The bit fields of the BSCR are shown in Figure 3-7 and are described in Table 3-5.

| 15 | 14–13 | 12 | 11–8 |
|---|---|---|---|
| CONSEC | DIVFCT | IACKOFF | Reserved |
| R/W-1 | R/W-11 | R/W-1 | R |

| 7–3 | 2 | 1 | 0 |
|---|---|---|---|
| Reserved | HBH | BH | Reserved |
| R | R/W-0 | R/W-0 | R |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-7. Bank-Switching Control Register BSCR)[MMR Address 0029h]**

**Table 3-5. Bank-Switching Control Register (BSCR) Fields**

| BIT | NAME | RESET VALUE | FUNCTION |
|---|---|---|---|
| 15 | $\overline{\text{CONSEC}}$ (1) | 1 | Consecutive bank-switching**.** Specifies the bank-switching mode.<br>• $\overline{\text{CONSEC}}$ = 0: Bank-switching on 32K bank boundaries only. This bit is cleared if fast access is desired for continuous memory reads (i.e., no starting and trailing cycles between read cycles).<br>• $\overline{\text{CONSEC}}$ = 1: Consecutive bank switches on external memory reads. Each read cycle consists of 3 cycles: starting cycle, read cycle, and trailing cycle. |
| 14-13 | DIVFCT | 11 | CLKOUT output divide factor**.** The CLKOUT output is driven by an on-chip source having a frequency equal to 1/(DIVFCT+1) of the DSP clock.<br>• DIVFCT = 00: CLKOUT is not divided.<br>• DIVFCT = 01: CLKOUT is divided by 2 from the DSP clock.<br>• DIVFCT = 10: CLKOUT is divided by 3 from the DSP clock.<br>• DIVFCT = 11: CLKOUT is divided by 4 from the DSP clock (default value following reset). |
| 12 | IACKOFF | 11 | $\overline{\text{IACK}}$ signal output off**.** Controls the output of the $\overline{\text{IACK}}$ signal. IACKOFF is set to 1 at reset.<br>• IACKOFF = 0: The $\overline{\text{IACK}}$ signal output off function is disabled.<br>• IACKOFF = 1: The $\overline{\text{IACK}}$ signal output off function is enabled. |
| 11-3 | Rsvd | | Reserved |
| 2 | HBH | 0 | HPI bus holder. Controls the HPI bus holder. HBH is cleared to 0 at reset.<br>• HBH = 0: The bus holder is disabled except when HPI16 = 1.<br>• HBH = 1: The bus holder is enabled. When not driven, the HPI data bus, HD[7:0] is held in the previous logic level. |
| 1 | BH | 0 | Bus holder. Controls the bus holder. BH is cleared to 0 at reset.<br>• BH = 0: The bus holder is disabled.<br>• BH = 1: The bus holder is enabled. When not driven, the data bus, D[15:0] is held in the previous logic level. |
| 0 | Rsvd | | Reserved |

(1) For additional information, see Section 3.11 of this document.

The device has an internal register that holds the MSB of the last address used for a read or write operation in program or data space. In the non-consecutive bank switches ($\overline{\text{CONSEC}}$ = 0), if the MSB of the address used for the current read does not match that contained in this internal register, the $\overline{\text{MSTRB}}$ (memory strobe) signal is not asserted for one CLKOUT cycle. During this extra cycle, the address bus switches to the new address. The contents of the internal register are replaced with the MSB for the read of the current address. If the MSB of the address used for the current read matches the bits in the register, a normal read cycle occurs.

In non-consecutive bank switches ($\overline{\text{CONSEC}}$ = 0), if repeated reads are performed from the same memory bank, no extra cycles are inserted. When a read is performed from a different memory bank, memory conflicts are avoided by inserting an extra cycle. For more information, see Section 3.11 of this document.

The bank-switching mechanism automatically inserts one extra cycle in the following cases:

- A memory read followed by another memory read from a different memory bank.
- A program-memory read followed by a data-memory read.
- A data-memory read followed by a program-memory read.
- A program-memory read followed by another program-memory read from a different page.

### *3.6.3 Bus Holders*

The device has two bus holder control bits, BH (BSCR[1]) and HBH (BSCR[2]), to control the bus keepers of the address bus (A[17-0]), data bus (D[15-0]), and the HPI data bus (HD[7-0]). Bus keeper enabling/disabling is described in Table 3-6.

**Table 3-6. Bus Holder Control Bits**

| HPI16 PIN | BH | HBH | D[15-0] | A[17-0] | HD[7-0] |
|---|---|---|---|---|---|
| 0 | 0 | 0 | OFF | OFF | OFF |
| 0 | 0 | 1 | OFF | OFF | ON |
| 0 | 1 | 0 | ON | OFF | OFF |
| 0 | 1 | 1 | ON | OFF | ON |
| 1 | 0 | 0 | OFF | OFF | ON |
| 1 | 0 | 1 | OFF | ON | ON |
| 1 | 1 | 0 | ON | OFF | ON |
| 1 | 1 | 1 | ON | ON | ON |

## 3.7 Parallel I/O Ports

The device has a total of 64K I/O ports. These ports can be addressed by the PORTR instruction or the PORTW instruction. The $\overline{\text{IS}}$ signal indicates a read/write operation through an I/O port. The device can interface easily with external devices through the I/O ports while requiring minimal off-chip address-decoding circuits.

### 3.7.1 Enhanced 8-/16-Bit Host-Port Interface (HPI8/16)

The host-port interface, also referred to as the HPI8/16, is an enhanced version of the standard 8-bit HPI found on earlier TMS320C54x™ DSPs (542, 545, 548, and 549). The HPI can be used to interface to an 8-bit or 16-bit host. When the address and data buses for external I/O is not used (to interface to external devices in program/data/IO spaces), the HPI can be configured as an HPI16 to interface to a 16-bit host. This configuration can be accomplished by connecting the HPI16 pin to logic "1”.

When the HPI16 pin is connected to a logic „0”, the HPI is configured as an HPI8. The HPI8 is an 8-bit parallel port for interprocessor communication. The features of the HPI8 include:

**Standard features:**

- Sequential transfers (with autoincrement) or random-access transfers
- Host interrupt and C54x™ interrupt capability
- Multiple data strobes and control pins for interface flexibility

The HPI8 interface consists of an 8-bit bidirectional data bus and various control signals. Sixteen-bit transfers are accomplished in two parts with the HBIL input designating high or low byte. The host communicates with the HPI8 through three dedicated registers — the HPI address register (HPIA), the HPI data register (HPID), and the HPI control register (HPIC). The HPIA and HPID registers are only accessible by the host, and the HPIC register is accessible by both the host and the device.

**Enhanced features:**

- Access to entire on-chip RAM through DMA bus
- Capability to continue transferring during emulation stop

The HPI16 is an enhanced 16-bit version of the TMS320C54x™ DSP 8-bit host-port interface (HPI8). The HPI16 is designed to allow a 16-bit host to access the DSP on-chip memory, with the host acting as the master of the interface. Some of the features of the HPI16 include:

- 16-bit bidirectional data bus
- Multiple data strobes and control signals to allow glueless interfacing to a variety of hosts
- Only nonmultiplexed address/data modes are supported
- 18-bit address bus used in nonmultiplexed mode to allow access to all internal memory (including internal extended address pages)
- HRDY signal to hold off host accesses due to DMA latency
- The HPI16 acts as a slave to a 16-bit host processor and allows access to the on-chip memory of the DSP.

**NOTE**

Only the nonmultiplexed mode is supported when the HPI is configured as a HPI16.

The HPI functions as a slave and enables the host processor to access the on-chip memory. A major enhancement to the HPI over previous versions is that it allows host access to the entire on-chip memory range of the DSP. The host and the DSP both have access to the on-chip RAM at all times and host accesses are always synchronized to the DSP clock. If the host and the DSP contend for access to the same location, the host has priority, and the DSP waits for one cycle. Note that since host accesses are always synchronized to the device clock, an active input clock (CLKIN) is required for HPI accesses during IDLE states, and host accesses are not allowed while the device reset pin is asserted.

### 3.7.2 HPI Nonmultiplexed Mode

In *nonmultiplexed* mode, a host with separate address/data buses can access the HPI16 data register (HPID) via the HD 16-bit bidirectional data bus, and the address register (HPIA) via the 18-bit HA address bus. The host initiates the access with the strobe signals ($\overline{\text{HDS1}}$, $\overline{\text{HDS2}}$, $\overline{\text{HCS}}$) and controls the direction of the access with the HR/$\overline{\text{W}}$ signal. The HPI16 can stall host accesses via the HRDY signal. Note that the HPIC register is not available in *nonmultiplexed* mode since there are no HCNTL signals available. All host accesses initiate a DMA read or write access. Figure 3-8 shows a block diagram of the HPI16 in *nonmultiplexed* mode. Figure 3-9 shows the HPI memory map.

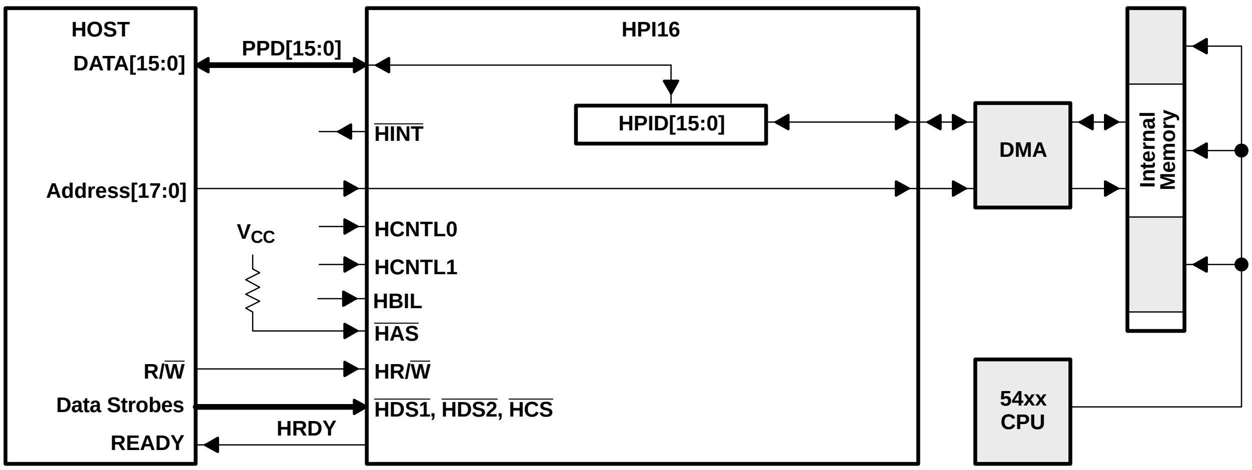



**Figure 3-8. Host-Port Interface — Nonmulltiplexed Mode**

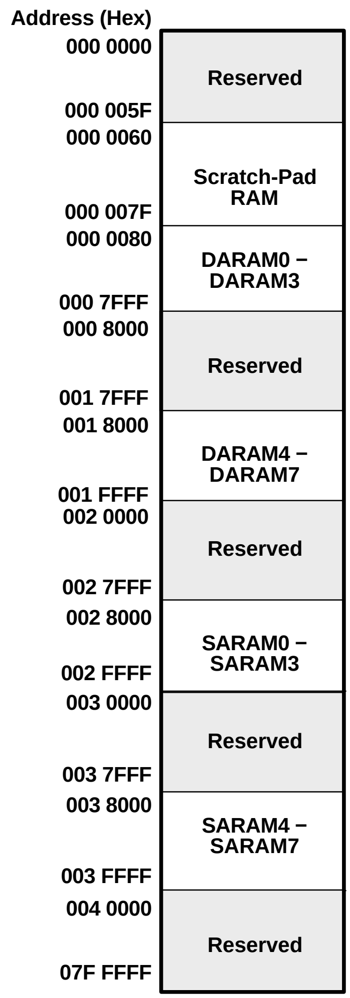



**Figure 3-9. HPI Memory Map**

## 3.8 Multichannel Buffered Serial Ports (McBSPs)

The device provides three high-speed, full-duplex, multichannel buffered serial ports that allow direct interface to other C54x/LC54x devices, codecs, and other devices in a system. The McBSPs are based on the standard serial-port interface found on other 54x devices. Like their predecessors, the McBSPs provide:

- Full-duplex communication
- Double-buffer data registers, which allow a continuous data stream
- Independent framing and clocking for receive and transmit

In addition, the McBSPs have the following capabilities:

- Direct interface to:
  - T1/E1 framers
  - MVIP switching compatible and ST-BUS compliant devices
  - IOM-2 compliant devices
  - AC97-compliant devices
  - IIS-compliant devices
  - Serial peripheral interface
- Multichannel transmit and receive of up to 128 channels
- A wide selection of data sizes, including 8, 12, 16, 20, 24, or 32 bits
- µ-law and A-law companding
- Programmable polarity for both frame synchronization and data clocks
- Programmable internal clock and frame generation

The McBSP consists of a data path and control path. The six pins, BDX, BDR, BFSX, BFSR, BCLKX, and BCLKR, connect the control and data paths to external devices. The implemented pins can be programmed as general-purpose I/O pins if they are not used for serial communication.

The data is communicated to devices interfacing to the McBSP by way of the data transmit (BDX) pin for transmit and the data receive (BDR) pin for receive. The CPU or DMA reads the received data from the data receive register (DRR) and writes the data to be transmitted to the data transmit register (DXR). Data written to the DXR is shifted out to BDX by way of the transmit shift register (XSR). Similarly, receive data on the BDR pin is shifted into the receive shift register (RSR) and copied into the receive buffer register (RBR). RBR is then copied to DRR, which can be read by the CPU or DMA. This allows internal data movement and external data communications simultaneously.

Control information in the form of clocking and frame synchronization is communicated by way of BCLKX, BCLKR, BFSX, and BFSR. The device communicates to the McBSP by way of 16-bit-wide control registers accessible via the internal peripheral bus.

The control block consists of internal clock generation, frame synchronization signal generation, and their control, and multichannel selection. This control block sends notification of important events to the CPU and DMA by way of two interrupt signals, XINT and RINT, and two event signals, XEVT and REVT.

The on-chip companding hardware allows compression and expansion of data in either µ-law or A-law format. When companding is used, transmitted data is encoded according to the specified companding law and received data is decoded to 2s complement format.

The sample rate generator provides the McBSP with several means of selecting clocking and framing for both the receiver and transmitter. Both the receiver and transmitter can select clocking and framing independently.

The McBSP allows the multiple channels to be independently selected for the transmitter and receiver. When multiple channels are selected, each frame represents a time-division multiplexed (TDM) data stream. In using time-division multiplexed data streams, the CPU may only need to process a few of them. Thus, to save memory and bus bandwidth, multichannel selection allows independent enabling of particular channels for transmission and reception. All 128 channels in a bit stream consisting of a maximum of 128 channels can be enabled.

| 15–10 | 9 | 8 |
|---|---|---|
| Reserved | XMCME | XPBBLK |
| R | R/W | R/W |

| 7 | 6–5 | 4–2 | 1 | 0 |
|---|---|---|---|---|
| XPBBLK | XPABLK | XCBLK | Reserved | XMCM |
| R/W | R/W | R | R | R/W |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-10. Multichannel Control Register (MCR1)**

| 15–10 | 9 | 8 |
|---|---|---|
| Reserved | RMCME | RPBBLK |
| R | R/W | R/W |

| 7 | 6–5 | 4–2 | 1 | 0 |
|---|---|---|---|---|
| RPBBLK | RPABLK | RCBLK | Reserved | RMCM |
| R/W | R/W | R | R | R/W |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-11. Multichannel Control Register (MCR2)**

The McBSP has two working modes:

- In the first mode, when (R/X)MCME = 0, it is comparable with the McBSPs used in the 5410 where the normal 32-channel selection is enabled (default).
- In the second mode, when (R/X)MCME = 1, it has 128-channel selection capability. Multichannel control register Bit 9, (R/X)MCME, is used as the 128-channel selection enable bit. Once (R/X)MCME = 1, twelve new registers ((R/X)CERC - (R/X)CERH) are used to enable the 128-channel selection.

The clock stop mode (CLKSTP) in the McBSP provides compatibility with the serial port interface protocol. Clock stop mode works with only single-phase frames and one word per frame. The word sizes supported by the McBSP are programmable for 8-, 12-, 16-, 20-, 24-, or 32-bit operation. When the McBSP is configured to operate in SPI mode, both the transmitter and the receiver operate together as a master or as a slave.

Although the BCLKS pin is not available on the device PGE and GGU packages, the device is capable of synchronization to external clock sources. BCLKX or BCLKR can be used by the sample rate generator for external synchronization. The sample rate clock mode extended (SCLKME) bit field is located in the PCR to accommodate this option.

| 15–14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|
| Reserved | XIOEN | RIOEN | FSXM | FSRM | CLKXM | CLKRM |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SCLKME | CLKS STAT | DX STAT | DR STAT | FSXP | FSRP | CLKXP | CLKRP |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-12. Pin Control Register (PCR)**

The selection of sample rate input clock is made by the combination of the CLKSM (bit 13 in SRGR2) bit value and the SCLKME bit value as shown in Table 3-7.

**Table 3-7. Sample Rate Input Clock Selection**

| SCLKME | CLKSM | SAMPLE RATE CLOCK MODE |
|---|---|---|
| 0 | 0 | Reserved (CLKS pin unavailable) |
| 0 | 1 | CPU clock |
| 1 | 0 | BCLKR |
| 1 | 1 | BCLKX |

When the SCLKME bit is cleared to 0, the CLKSM bit is used, as before, to select either the CPU clock or the CLKS pin (not bonded out on the device package) as the sample rate input clock. Setting the SCLKME bit to 1 enables the CLKSM bit to select between the BCLKR pin or BCLKX pin for the sample rate input clock.

When either the BCLKR or CLKX is configured this way, the output buffer for the selected pin is automatically disabled. For example, with SCLKME = 1 and CLKSM = 0, the BCLKR pin is configured as the input of the sample rate generator. Both the transmitter and receiver circuits can be synchronized to the sample rate generator output by setting the CLKXM and CLKRM bits of the pin configuration register (PCR) to 1. Note that the sample rate generator output will only be driven on the BCLKX pin since the BCLKR output buffer is automatically disabled.

The McBSP is fully static and operates at arbitrary low clock frequencies. For maximum operating frequency, see Section 5.5.10.

## 3.9 Hardware Timer

The device features a 16-bit timing circuit with a 4-bit prescaler. The timer counter is decremented by one every CLKOUT cycle. Each time the counter decrements to 0, a timer interrupt is generated. The timer can be stopped, restarted, reset, or disabled by specific status bits.

## 3.10 Clock Generator

The clock generator provides clocks to the device, and consists of a phase-locked loop (PLL) circuit. The clock generator requires a reference clock input, which can be provided from an external clock source. The reference clock input is then divided by two (DIV mode) to generate clocks for the device, or the PLL circuit can be used (PLL mode) to generate the device clock by multiplying the reference clock frequency by a scale factor, allowing use of a clock source with a lower frequency than that of the CPU. The PLL is an adaptive circuit that, once synchronized, locks onto and tracks an input clock signal.

When the PLL is initially started, it enters a transitional mode during which the PLL acquires lock with the input signal. Once the PLL is locked, it continues to track and maintain synchronization with the input signal. Then, other internal clock circuitry allows the synthesis of new clock frequencies for use as master clock for the device.

This clock generator allows system designers to select the clock source. The sources that drive the clock generator are:

- A crystal resonator circuit. The crystal resonator circuit is connected across the X1 and X2/CLKIN pins of the device to enable the internal oscillator.
- An external clock. The external clock source is directly connected to the X2/CLKIN pin, and X1 is left unconnected.

**NOTE**

The crystal oscillator function is not supported by all die revisions of the device. See the *TMS320VC5416 Digital Signal Processor Silicon Errata* (literature number SPRZ172) to verify which die revisions support this functionality.

The software-programmable PLL features a high level of flexibility, and includes a clock scaler that provides various clock multiplier ratios, capability to directly enable and disable the PLL, and a PLL lock timer that can be used to delay switching to PLL clocking mode of the device until lock is achieved. Devices that have a built-in software-programmable PLL can be configured in one of two clock modes:

- PLL mode. The input clock (X2/CLKIN) is multiplied by 1 of 31 possible ratios.
- DIV (divider) mode. The input clock is divided by 2 or 4. Note that when DIV mode is used, the PLL can be completely disabled in order to minimize power dissipation.

The software-programmable PLL is controlled using the 16-bit memory-mapped (address 0058h) clock mode register (CLKMD). The CLKMD register is used to define the clock configuration of the PLL clock module. Note that upon reset, the CLKMD register is initialized with a predetermined value dependent only upon the state of the CLKMD1 - CLKMD3 pins. For more programming information, see the *TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals* (literature number SPRU131). The CLKMD pin configured clock options are shown in Table 3-8.

**Table 3-8. Clock Mode Settings at Reset**

| CLKMD1 | CLKMD2 | CLKMD3 | CLKMD RESET VALUE | CLOCK MODE(1) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0000h | 1/2 (PLL disabled) |
| 0 | 0 | 1 | 9007h | PLL x 10 |
| 0 | 1 | 0 | 4007h | PLL x 5 |
| 1 | 0 | 0 | 1007h | PLL x 2 |
| 1 | 1 | 0 | F007h | PLL x 1 |
| 1 | 1 | 1 | 0000h | 1/2 (PLL disabled) |
| 1 | 0 | 1 | F000h | 1/4 (PLL disabled) |
| 0 | 1 | 1 | — | Reserved (Bypass mode) |

(1) The external CLKMD1-CLKMD3 pins are sampled to determine the desired clock generation mode while $\overline{RS}$ is low. Following reset, the clock generation mode can be reconfigured by writing to the internal clock mode register in software.

## 3.11 Enhanced External Parallel Interface (XIO2)

The device external interface has been redesigned to include several improvements, including: simplification of the bus sequence, more immunity to bus contention when transitioning between read and write operations, the ability for external memory access to the DMA controller, and optimization of the power-down modes.

The bus sequence on the device still maintains all of the same interface signals as on previous 54x devices, but the signal sequence has been simplified. Most external accesses now require 3 cycles composed of a leading cycle, an active (read or write) cycle, and a trailing cycle. The leading and trailing cycles provide additional immunity against bus contention when switching between read operations and write operations. To maintain high-speed read access, a consecutive read mode that performs single-cycle reads as on previous 54x devices is available.

Figure 3-13 shows the bus sequence for three cases: all I/O reads, memory reads in nonconsecutive mode, or single memory reads in consecutive mode. The accesses shown in Figure 3-13 always require 3 CLKOUT cycles to complete.

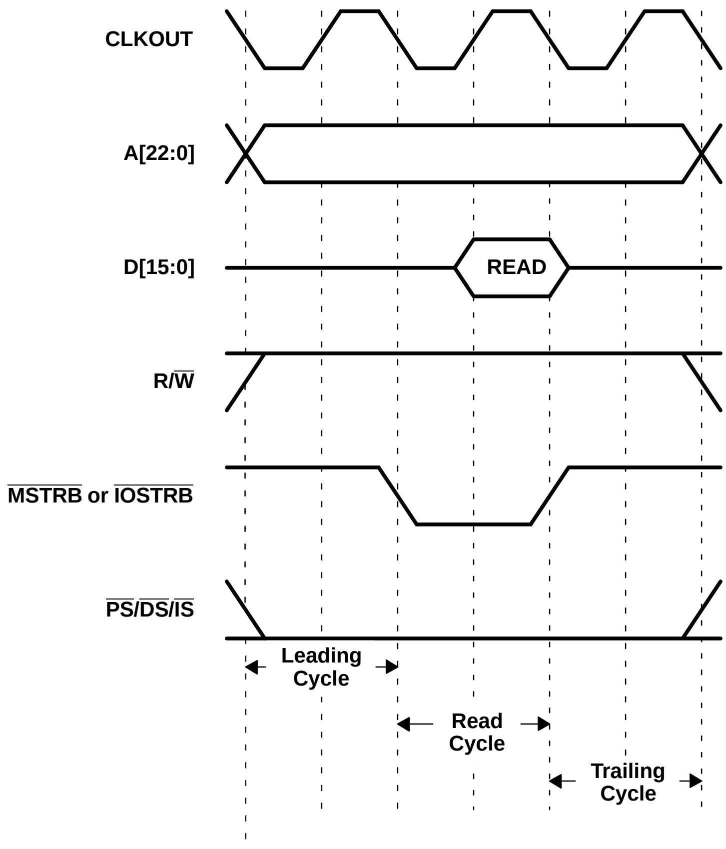



**Figure 3-13. Nonconsecutive Memory Read and I/O Read Bus Sequence**

Figure 3-14 shows the bus sequence for repeated memory reads in consecutive mode. The accesses shown in Figure 3-14 require (2 + n) CLKOUT cycles to complete, where n is the number of consecutive reads performed.

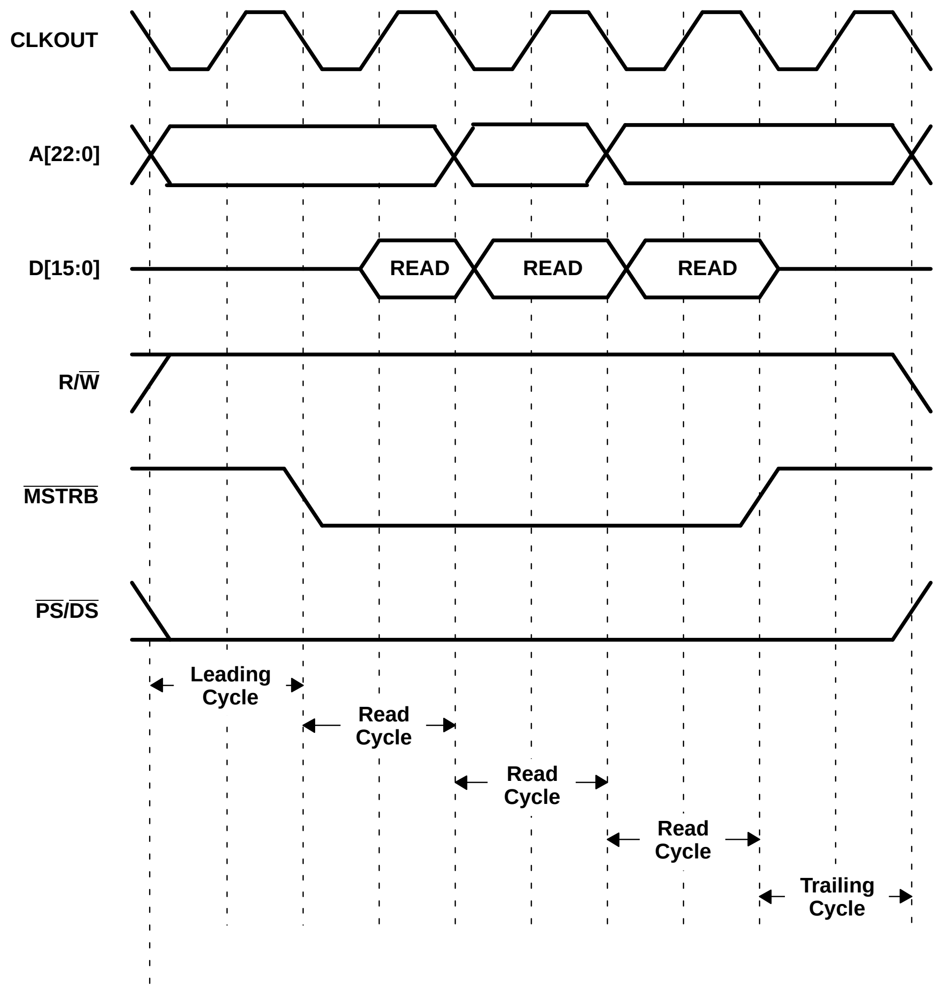



**Figure 3-14. Consecutive Memory Read Bus Sequence (n = 3 reads)**

Figure 3-15 shows the bus sequence for all memory writes and I/O writes. The accesses shown in Figure 3-15 always require 3 CLKOUT cycles to complete.

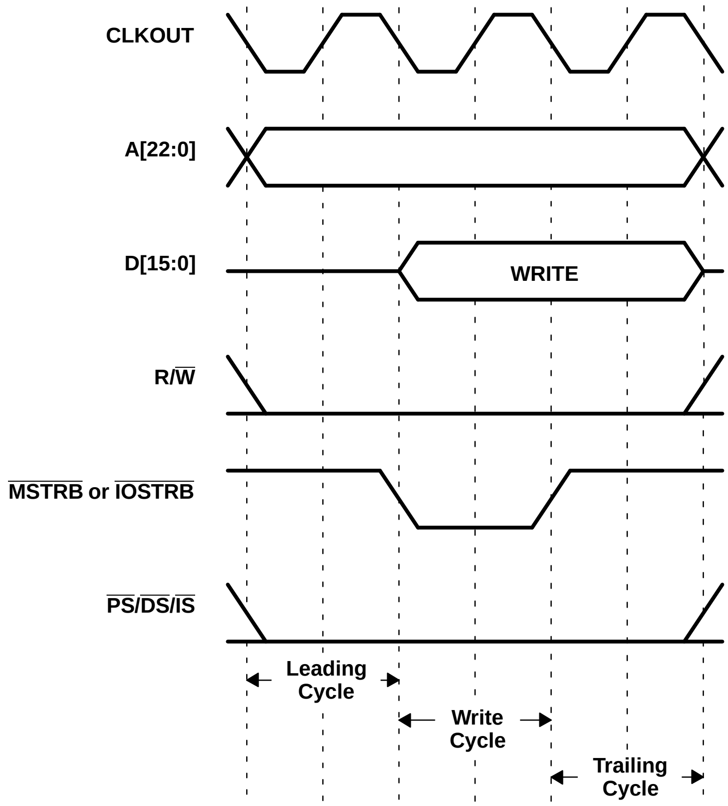



**Figure 3-15. Memory Write and I/O Write Bus Sequence**

The enhanced interface also provides the ability for DMA transfers to extend to external memory. For more information on DMA capability, see the DMA sections that follow.

The enhanced interface improves the low-power performance already present on the TMS320C5000™ DSP platform by switching off the internal clocks to the interface when it is not being used. This power-saving feature is automatic, requires no software setup, and causes no latency in the operation of the interface.

Additional features integrated in the enhanced interface are the ability to automatically insert bank-switching cycles when crossing 32K memory boundaries (see Section 3.6.2), the ability to program up to 14 wait states through software (see Section 3.6.1), and the ability to divide down CLKOUT by a factor of 1, 2, 3, or 4. Dividing down CLKOUT provides an alternative to wait states when interfacing to slower external memory or peripheral devices. While inserting wait states extends the bus sequence during read or write accesses, it does not slow down the bus signal sequences at the beginning and the end of the access. Dividing down CLKOUT provides a method of slowing the entire bus sequence when necessary. The CLKOUT divide-down factor is controlled through the DIVFCT field in the bank-switching control register (BSCR) (see Table 3-5).

## 3.12 DMA Controller

The device direct memory access (DMA) controller transfers data between points in the memory map without intervention by the CPU. The DMA allows movements of data to and from internal program/data memory, internal peripherals (such as the McBSPs), or external memory devices to occur in the background of CPU operation. The DMA has six independent programmable channels, allowing six different contexts for DMA operation.

### *3.12.1 Features*

The DMA has the following features:

- The DMA operates independently of the CPU.
- The DMA has six channels. The DMA can keep track of the contexts of six independent block transfers.
- The DMA has higher priority than the CPU for both internal and external accesses.
- Each channel has independently programmable priorities.
- Each channel's source and destination address registers can have configurable indexes through memory on each read and write transfer, respectively. The address may remain constant, be post-incremented, be post-decremented, or be adjusted by a programmable value.
- Each read or write internal transfer may be initialized by selected events.
- On completion of a half- or entire-block transfer, each DMA channel may send an interrupt to the CPU.
- The DMA can perform double-word internal transfers (a 32-bit transfer of two 16-bit words).

### *3.12.2 DMA External Access*

The DMA supports external accesses to data, I/O, and extended program memory. These overlay pages are only visible to the DMA controller. A maximum of two DMA channels can be used for external memory accesses. The DMA external accesses require a minimum of 8 cycles for external writes and a minimum of 11 cycles for external reads assuming the XIO02 is in consecutive mode ($\overline{\text{CONSEC}}$ = 1), wait state is set to two, and CLKOUT is not divided (DIVFCT = 00).

The control of the bus is arbitrated between the CPU and the DMA. While the DMA or CPU is in control of the external bus, the other will be held-off via wait states until the current transfer is complete. The DMA takes precedence over XIO requests.

- Only two channels are available for external accesses. (One for external reads and one for external writes.)
- Single-word (16-bit) transfers are supported for external accesses.
- The DMA does not support transfers from the peripherals to external memory.
- The DMA does not support transfers from external memory to the peripherals.
- The DMA does not support external-to-external accesses.
- The DMA does not support synchronized external accesses.

| 15 | 14 | 13 | 12 | 11 | 10 | | 8 |
|---|---|---|---|---|---|---|---|
| AUTOINIT | DINM | IMOD | CTMOD | SLAXS | SIND | | |

| 7 | 6 | 5 | 4 | | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DMS | | DLAXS | DIND | | | DMD | |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-16. DMA Transfer Mode Control Register (DMMCRn)**

These new bit fields were created to allow the user to define the space-select for the DMA (internal/external). The functions of the DLAXS and SLAXS bits are as follows:

- DLAXS(DMMCRn[5]) Destination
  - 0 = No external access (default internal)
  - 1 = External access
- SLAXS(DMMCRn[11]) Source
  - 0 = No external access (default internal)
  - 1 = External access

Table 3-9 lists the DMD bit values and their corresponding destination space.

**Table 3-9. DMD Section of the DMMCRn Register**

| DMD | Destination Space |
|---|---|
| 00 | PS |
| 01 | DS |
| 10 | I/O |
| 11 | Reserved |

For the CPU external access, software can configure the memory cells to reside inside or outside the program address map. When the cells are mapped into program space, the device automatically accesses them when their addresses are within bounds. When the address generation logic generates an address outside its bounds, the device automatically generates an external access.

### *3.12.3 DMA Memory Maps*

The DMA memory maps, shown in Figure 3-17 and Figure 3-18, allows the DMA transfer to be unaffected by the status of the MP/$\overline{MC}$, DROM, and OVLY bits.

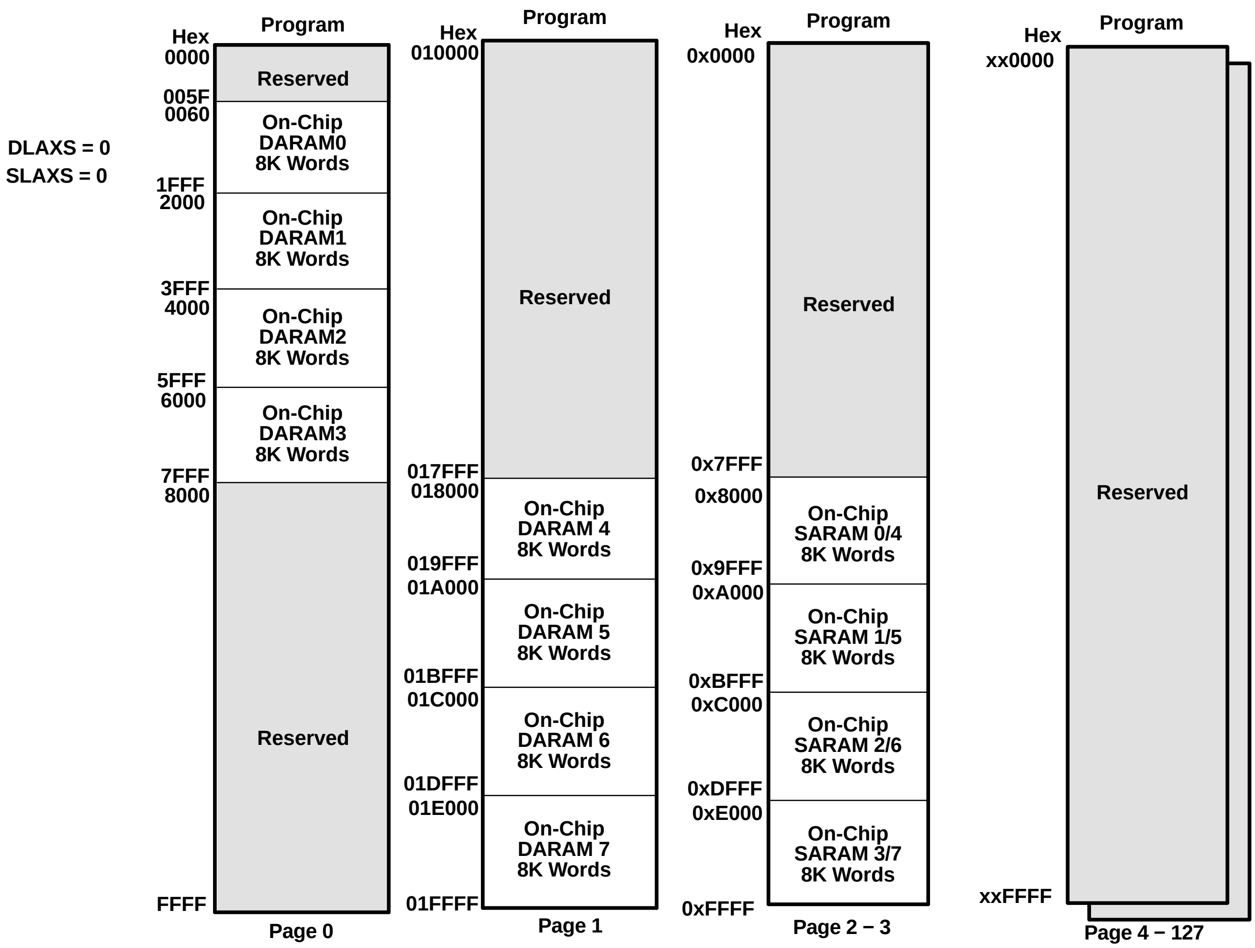



**Figure 3-17. On-Chip DMA Memory Map for Program Space (DLAXS = 0 and SLAXS = 0)**

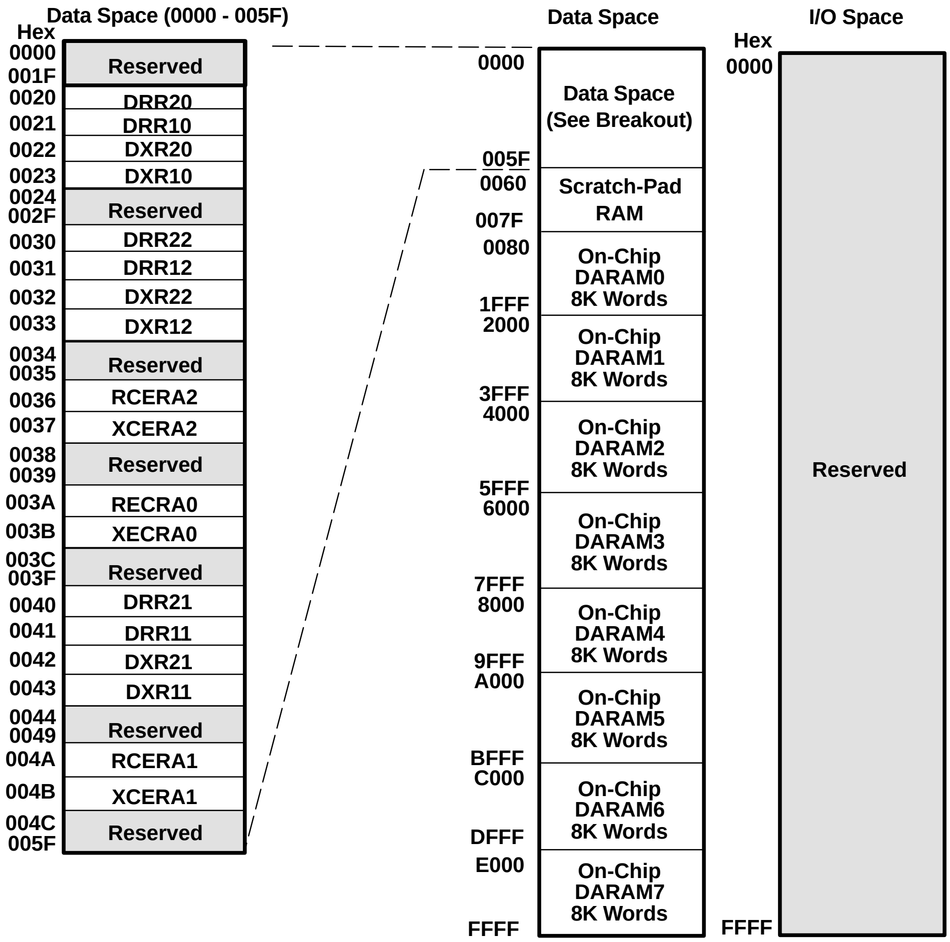



**Figure 3-18. On-Chip DMA Memory Map for Data and IO Space (DLAXS = 0 and SLAXS = 0)**

### 3.12.4 DMA Priority Level

Each DMA channel can be independently assigned high- or low-priority relative to each other. Multiple DMA channels that are assigned to the same priority level are handled in a round-robin manner.

### 3.12.5 DMA Source/Destination Address Modification

The DMA provides flexible address-indexing modes for easy implementation of data management schemes such as autobuffering and circular buffers. Source and destination addresses can be indexed separately and can be post-incremented, post-decremented, or post-incremented with a specified index offset.

### 3.12.6 DMA in Autoinitialization Mode

The DMA can automatically reinitialize itself after completion of a block transfer. Some of the DMA registers can be preloaded for the next block transfer through the DMA reload registers (DMGSA, DMGDA, DMGCR, and DMGFR). Autoinitialization allows:

- Continuous operation: Normally, the CPU would have to reinitialize the DMA immediately after the completion of the current block transfers, but with the reload registers, it can reinitialize these values for the next block transfer any time after the current block transfer begins.
- Repetitive operation:The CPU does not preload the reload register with new values for each block transfer but only loads them on the first block transfer.

The DMA has been enhanced to expand the DMA reload register sets. Each DMA channel now has its own DMA reload register set. For example, the DMA reload register set for channel 0 has DMGSA0, DMGDA0, DMGCR0, and DMGFR0 while DMA channel 1 has DMGSA1, DMGDA1, DMGCR1, and DMGFR1, etc.

To utilize the additional DMA reload registers, the AUTOIX bit is added to the DMPREC register as shown in Figure 3-19.

| 15 | 14 | 13–8 |
|---|---|---|
| FREE | AUTOIX | DPRC[5:0] |

| 7–6 | 5–0 |
|---|---|
| IOSEL | DE[5:0] |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-19. DMPREC Register**

**Table 3-10. DMA Reload Register Selection**

| AUTOIX | DMA RELOAD REGISTER USAGE IN AUTO INIT MODE |
|---|---|
| 0 (default) | All DMA channels use DMGSA0, DMGDA0, DMGCR0 and DMGFR0 |
| 1 | Each DMA channel uses its own set of reload registers |

### 3.12.7 DMA Transfer Counting

The DMA channel element count register (DMCTRx) and the frame count register (DMFRCx) contain bit fields that represent the number of frames and the number of elements per frame to be transferred.

- Frame count. This 8-bit value defines the total number of frames in the block transfer. The maximum number of frames per block transfer is 128 (FRAME COUNT= 0FFh). The counter is decremented upon the last read transfer in a frame transfer. Once the last frame is transferred, the selected 8-bit counter is reloaded with the DMA global frame reload register (DMGFR) if the AUTOINIT bit is set to 1. A frame count of 0 (default value) means the block transfer contains a single frame.
- Element count. This 16-bit value defines the number of elements per frame. This counter is decremented after the read transfer of each element. The maximum number of elements per frame is 65536 (DMCTRn = 0FFFFh). In autoinitialization mode, once the last frame is transferred, the counter is reloaded with the DMA global count reload register (DMGCR).

### 3.12.8 DMA Transfer in Doubleword Mode

Doubleword mode allows the DMA to transfer 32-bit words in any index mode. In doubleword mode, two consecutive 16-bit transfers are initiated and the source and destination addresses are automatically updated following each transfer. In this mode, each 32-bit word is considered to be one element.

### 3.12.9 DMA Channel Index Registers

The particular DMA channel index register is selected by way of the SIND and DIND fields in the DMA transfer mode control register (DMMCRn). Unlike basic address adjustment, in conjunction with the frame index DMFRI0 and DMFRI1, the DMA allows different adjustment amounts depending on whether or not the element transfer is the last in the current frame. The normal adjustment value (element index) is contained in the element index registers DMIDX0 and DMIDX1. The adjustment value (frame index) for the end of the frame, is determined by the selected DMA frame index register, either DMFRI0 or DMFRI1.

The element index and the frame index affect address adjustment as follows:

- Element index: For all except the last transfer in the frame, the element index determines the amount to be added to the DMA channel for the source/destination address register (DMSRCx/DMDSTx) as selected by the SIND/DIND bits.
- Frame index: If the transfer is the last in a frame, frame index is used for address adjustment as selected by the SIND/DIND bits. This occurs in both single-frame and multiframe transfers.

### 3.12.10 DMA Interrupts

The ability of the DMA to interrupt the CPU based on the status of the data transfer is configurable and is determined by the IMOD and DINM bits in the DMA transfer mode control register (DMMCRn). The available modes are shown in Table 3-11.

**Table 3-11. DMA Interrupts**

| MODE | DINM | IMOD | INTERRUPT |
|---|---|---|---|
| ABU (non-decrement) | 1 | 0 | At full buffer only |
| ABU (non-decrement) | 1 | 1 | At half buffer and full buffer |
| Multiframe | 1 | 0 | At block transfer complete (DMCTRn = DMSEFCn[7:0] = 0) |
| Multiframe | 1 | 1 | At end of frame and end of block (DMCTRn = 0) |
| Either | 0 | X | No interrupt generated |
| Either | 0 | X | No interrupt generated |

### 3.12.11 DMA Controller Synchronization Events

The transfers associated with each DMA channel can be synchronized to one of several events. The DSYN bit field of the DMSEFCn register selects the synchronization event for a channel. The list of possible events and the DSYN values are shown in Table 3-12.

**Table 3-12. DMA Synchronization Events**

| DSYN VALUE | DMA SYNCHRONIZATION EVENT |
|---|---|
| 0000b | No synchronization used |
| 0001b | McBSP0 receive event |
| 0010b | McBSP0 transmit event |
| 0011b | McBSP2 receive event |
| 0100b | McBSP2 transmit event |

**Table 3-12. DMA Synchronization Events (continued)**

| DSYN VALUE | DMA SYNCHRONIZATION EVENT |
|---|---|
| 0101b | McBSP1 receive event |
| 0110b | McBSP1 transmit event |
| 0111b | McBSP0 receive event - ABIS mode |
| 1000b | McBSP0 transmit event - ABIS mode |
| 1001b | McBSP2 receive event - ABIS mode |
| 1010b | McBSP2 transmit event - ABIS mode |
| 1011b | McBSP1 receive event - ABIS mode |
| 1100b | McBSP1 transmit event - ABIS mode |
| 1101b | Timer interrupt event |
| 1110b | $\overline{\text{INT3}}$ goes active |
| 1111b | Reserved |

The DMA controller can generate a CPU interrupt for each of the six channels. However, due to a limit on the number of internal CPU interrupt inputs, channels 0, 1, 2, and 3 are multiplexed with other interrupt sources. DMA channels 0, 1, 2, and 3 share an interrupt line with the receive and transmit portions of the McBSP. When the device is reset, the interrupts from these three DMA channels are deselected. The INTSEL bit field in the DMPREC register can be used to select these interrupts, as shown in Table 3-13.

**Table 3-13. DMA Channel Interrupt Selection**

| INTSEL Value | IMR/IFR[6] | IMR/IFR[7] | IMR/IFR[10] | IMR/IFR[11] |
|---|---|---|---|---|
| 00b (reset) | BRINT2 | BXINT2 | BRINT1 | BXINT1 |
| 01b | BRINT2 | BXINT2 | DMAC2 | DMAC3 |
| 10b | DMAC0 | DMAC1 | DMAC2 | DMAC3 |
| 11b | Reserved | | | |

## 3.13 General-Purpose I/O Pins

In addition to the standard $\overline{\text{BIO}}$ and XF pins, the device has pins that can be configured for general-purpose I/O. These pins are:

- 18 McBSP pins
  - BCLKX0/1/2,
  - BCLKR0/1/2
  - BDR0/1/2
  - BFSX0/1/2
  - BFSR0/1/2
  - BDX0/1/2
- 8 HPI data pins
  - HD0-HD7

The general-purpose I/O function of these pins is only available when the primary pin function is not required.

### *3.13.1 McBSP Pins as General-Purpose I/O*

When the receive or transmit portion of a McBSP is in reset, its pins can be configured as general-purpose inputs or outputs. For more details on this feature, see Section 3.8.

### *3.13.2 HPI Data Pins as General-Purpose I/O*

The 8-bit bidirectional data bus of the HPI can be used as general-purpose input/output (GPIO) pins when the HPI is disabled (HPIENA = 0) or when the HPI is used in HPI16 mode (HPI16 = 1). Two memory-mapped registers are used to control the GPIO function of the HPI data pins—the general-purpose I/O control register (GPIOCR) and the general-purpose I/O status register (GPIOSR). The GPIOCR is shown in Figure 3-20.

| 15–8 |
|---|
| Reserved |
| 0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DIR7 | DIR6 | DIR5 | DIR4 | DIR3 | DIR2 | DIR1 | DIR0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-20. General-Purpose I/O Control Register (GPIOCR) [MMR Address 003Ch]**

The direction bits (DIRx) are used to configure HD0-HD7 as inputs or outputs.

The status of the GPIO pins can be monitored using the bits of the GPIOSR. The GPIOSR is shown in Figure 3-21.

| 15–8 |
|---|
| Reserved |
| 0 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| IO7 | IO6 | IO5 | IO4 | IO3 | IO2 | IO1 | IO0 |
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-21. General-Purpose I/O Status Register (GPIOSR) [MMR Address 003Dh]**

## 3.14 Device ID Register

A read-only memory-mapped register has been added to the device to allow user application software to identify on which device the program is being executed.

| 15–8 |
|---|
| CHIP ID |
| R |

| 7–4 | 3–0 |
|---|---|
| CHIP REVISION | SUBSYSID |
| R | R |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-22. Device ID Register (CSIDR) [MMR Address 003Eh]**

**Table 3-14. Device ID Register (CSIDR) Bits**

| BIT | FUNCTION |
|---|---|
| 15-8 | Chip ID (hex code of 16) |
| 7-4 | Chip revision ID |
| 3-0 | Subsystem ID (0000b for single core devices) |

## 3.15 Memory-Mapped Registers

The device has 27 memory-mapped CPU registers, which are mapped in data memory space address 0h to 1Fh. Each device also has a set of memory-mapped registers associated with peripherals. Table 3-15 gives a list of CPU memory-mapped registers (MMRs) available. Table 3-16 shows additional peripheral MMRs associated with the device.

**Table 3-15. CPU Memory-Mapped Registers**

| NAME | ADDRESS | | DESCRIPTION |
|---|---|---|---|
| | DEC | HEX | |
| IMR | 0 | 0 | Interrupt mask register |
| IFR | 1 | 1 | Interrupt flag register |
| — | 2-5 | 2-5 | Reserved for testing |
| ST0 | 6 | 6 | Status register 0 |
| ST1 | 7 | 7 | Status register 1 |
| AL | 8 | 8 | Accumulator A low word (15-0) |
| AH | 9 | 9 | Accumulator A high word (31-16) |
| AG | 10 | A | Accumulator A guard bits (39-32) |
| BL | 11 | B | Accumulator B low word (15-0) |
| BH | 12 | C | Accumulator B high word (31-16) |
| BG | 13 | D | Accumulator B guard bits (39-32) |
| TREG | 14 | E | Temporary register |
| TRN | 15 | F | Transition register |
| AR0 | 16 | 10 | Auxiliary register 0 |
| AR1 | 17 | 11 | Auxiliary register 1 |
| AR2 | 18 | 12 | Auxiliary register 2 |
| AR3 | 19 | 13 | Auxiliary register 3 |
| AR4 | 20 | 14 | Auxiliary register 4 |
| AR5 | 21 | 15 | Auxiliary register 5 |
| AR6 | 22 | 16 | Auxiliary register 6 |
| AR7 | 23 | 17 | Auxiliary register 7 |
| SP | 24 | 18 | Stack pointer register |
| BK | 25 | 19 | Circular buffer size register |
| BRC | 26 | 1A | Block repeat counter |
| RSA | 27 | 1B | Block repeat start address |
| REA | 28 | 1C | Block repeat end address |
| PMST | 29 | 1D | Processor mode status (PMST) register |
| XPC | 30 | 1E | Extended program page register |
| — | 31 | 1F | Reserved |

### Table 3-16. Peripheral Memory-Mapped Registers for Each DSP Subsystem

| NAME | ADDRESS | | DESCRIPTION |
|---|---|---|---|
| | DEC | HEX | |
| DRR20 | 32 | 20 | McBSP 0 Data Receive Register 2 |
| DRR10 | 33 | 21 | McBSP 0 Data Receive Register 1 |
| DXR20 | 34 | 22 | McBSP 0 Data Transmit Register 2 |
| DXR10 | 35 | 23 | McBSP 0 Data Transmit Register 1 |
| TIM | 36 | 24 | Timer Register |
| PRD | 37 | 25 | Timer Period Register |
| TCR | 38 | 26 | Timer Control Register |
| — | 39 | 27 | Reserved |
| SWWSR | 40 | 28 | Software Wait-State Register |
| BSCR | 41 | 29 | Bank-Switching Control Register |
| — | 42 | 2A | Reserved |
| SWCR | 43 | 2B | Software Wait-State Control Register |
| HPIC | 44 | 2C | HPI Control Register (HMODE = 0 only) |
| — | 45-47 | 2D-2F | Reserved |
| DRR22 | 48 | 30 | McBSP 2 Data Receive Register 2 |
| DRR12 | 49 | 31 | McBSP 2 Data Receive Register 1 |
| DXR22 | 50 | 32 | McBSP 2 Data Transmit Register 2 |
| DXR12 | 51 | 33 | McBSP 2 Data Transmit Register 1 |
| SPSA2 | 52 | 34 | McBSP 2 Subbank Address Register(1) |
| SPSD2 | 53 | 35 | McBSP 2 Subbank Data Register(1) |
| — | 54-55 | 36-37 | Reserved |
| SPSA0 | 56 | 38 | McBSP 0 Subbank Address Register(1) |
| SPSD0 | 57 | 39 | McBSP 0 Subbank Data Register(1) |
| — | 58-59 | 3A-3B | Reserved |
| GPIOCR | 60 | 3C | General-Purpose I/O Control Register |
| GPIOSR | 61 | 3D | General-Purpose I/O Status Register |
| CSIDR | 62 | 3E | Device ID Register |
| — | 63 | 3F | Reserved |
| DRR21 | 64 | 40 | McBSP 1 Data Receive Register 2 |
| DRR11 | 65 | 41 | McBSP 1 Data Receive Register 1 |
| DXR21 | 66 | 42 | McBSP 1 Data Transmit Register 2 |
| DXR11 | 67 | 43 | McBSP 1 Data Transmit Register 1 |
| — | 68-71 | 44-47 | Reserved |
| SPSA1 | 72 | 48 | McBSP 1 Subbank Address Register(1) |
| SPSD1 | 73 | 49 | McBSP 1 Subbank Data Register(1) |
| — | 74-83 | 4A-53 | Reserved |
| DMPREC | 84 | 54 | DMA Priority and Enable Control Register |
| DMSA | 85 | 55 | DMA Subbank Address Register(2) |
| DMSDI | 86 | 56 | DMA Subbank Data Register with Autoincrement(2) |
| DMSDN | 87 | 57 | DMA Subbank Data Register(2) |
| CLKMD | 88 | 58 | Clock Mode Register (CLKMD) |
| — | 89-95 | 59-5F | Reserved |

(1) See Table 3-17 for a detailed description of the McBSP control registers and their subaddresses.
(2) See Table 3-18 for a detailed description of the DMA subbank addressed registers.

## 3.16 McBSP Control Registers and Subaddresses

The control registers for the multichannel buffered serial port (McBSP) are accessed using the subbank addressing scheme. This allows a set or subbank of registers to be accessed through a single memory location. The McBSP subbank address register (SPSA) is used as a pointer to select a particular register within the subbank. The McBSP data register (SPSDx) is used to access (read or write) the selected register. Table 3-17 shows the McBSP control registers and their corresponding subaddresses.

**Table 3-17. McBSP Control Registers and Subaddresses**

| McBSP0 | | McBSP1 | | McBSP2 | | SUB- ADDRESS | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| NAME | ADDRESS | NAME | ADDRESS | NAME | ADDRESS | | |
| SPCR10 | 39h | SPCR11 | 49h | SPCR12 | 35h | 00h | Serial port control register 1 |
| SPCR20 | 39h | SPCR21 | 49h | SPCR22 | 35h | 01h | Serial port control register 2 |
| RCR10 | 39h | RCR11 | 49h | RCR12 | 35h | 02h | Receive control register 1 |
| RCR20 | 39h | RCR21 | 49h | RCR22 | 35h | 03h | Receive control register 2 |
| XCR10 | 39h | XCR11 | 49h | XCR12 | 35h | 04h | Transmit control register 1 |
| XCR20 | 39h | XCR21 | 49h | XCR22 | 35h | 05h | Transmit control register 2 |
| SRGR10 | 39h | SRGR11 | 49h | SRGR12 | 35h | 06h | Sample rate generator register 1 |
| SRGR20 | 39h | SRGR21 | 49h | SRGR22 | 35h | 07h | Sample rate generator register 2 |
| MCR10 | 39h | MCR11 | 49h | MCR12 | 35h | 08h | Multichannel control register 1 |
| MCR20 | 39h | MCR21 | 49h | MCR22 | 35h | 09h | Multichannel control register 2 |
| RCERA0 | 39h | RCERA1 | 49h | RCERA2 | 35h | 0Ah | Receive channel enable register partition A |
| RCERB0 | 39h | RCERB1 | 49h | RCERA2 | 35h | 0Bh | Receive channel enable register partition B |
| XCERA0 | 39h | XCERA1 | 49h | XCERA2 | 35h | 0Ch | Transmit channel enable register partition A |
| XCERB0 | 39h | XCERB1 | 49h | XCERA2 | 35h | 0Dh | Transmit channel enable register partition B |
| PCR0 | 39h | PCR1 | 49h | PCR2 | 35h | 0Eh | Pin control register |
| RCERC0 | 39h | RCERC1 | 49h | RCERC2 | 35h | 010h | Additional channel enable register for 128-channel selection |
| RCERD0 | 39h | RCERD1 | 49h | RCERD2 | 35h | 011h | Additional channel enable register for 128-channel selection |
| XCERC0 | 39h | XCERC1 | 49h | XCERC2 | 35h | 012h | Additional channel enable register for 128-channel selection |
| XCERD0 | 39h | XCERD1 | 49h | XCERD2 | 35h | 013h | Additional channel enable register for 128-channel selection |
| RCERE0 | 39h | RCERE1 | 49h | RCERE2 | 35h | 014h | Additional channel enable register for 128-channel selection |
| RCERF0 | 39h | RCERF1 | 49h | RCERF2 | 35h | 015h | Additional channel enable register for 128-channel selection |
| XCERE0 | 39h | XCERE1 | 49h | XCERE2 | 35h | 016h | Additional channel enable register for 128-channel selection |
| XCERF0 | 39h | XCERF1 | 49h | XCERF2 | 35h | 017h | Additional channel enable register for 128-channel selection |
| RCERG0 | 39h | RCERG1 | 49h | RCERG2 | 35h | 018h | Additional channel enable register for 128-channel selection |
| RCERH0 | 39h | RCERH1 | 49h | RCERH2 | 35h | 019h | Additional channel enable register for 128-channel selection |
| XCERG0 | 39h | XCERG1 | 49h | XCERG2 | 35h | 01Ah | Additional channel enable register for 128-channel selection |
| XCERH0 | 39h | XCERH1 | 49h | XCERH2 | 35h | 01Bh | Additional channel enable register for 128-channel selection |

## 3.17 DMA Subbank Addressed Registers

The direct memory access (DMA) controller has several control registers associated with it. The main control register (DMPREC) is a standard memory-mapped register. However, the other registers are accessed using the subbank addressing scheme. This allows a set or subbank of registers to be accessed through a single memory location. The DMA subbank address (DMSA) register is used as a pointer to select a particular register within the subbank, while the DMA subbank data (DMSD) register or the DMA subbank data register with autoincrement (DMSDI) is used to access (read or write) the selected register.

When the DMSDI register is used to access the subbank, the subbank address is automatically postincremented so that a subsequent access affects the next register within the subbank. This autoincrement feature is intended for efficient, successive accesses to several control registers. If the autoincrement feature is not required, the DMSDN register should be used to access the subbank. Table 3-18 shows the DMA controller subbank addressed registers and their corresponding subaddresses.

**Table 3-18. DMA Subbank Addressed Registers**

| NAME | ADDRESS | SUB- ADDRESS | DESCRIPTION |
|---|---|---|---|
| DMSRC0 | 56h/57h | 00h | DMA channel 0 source address register |
| DMDST0 | 56h/57h | 01h | DMA channel 0 destination address register |
| DMCTR0 | 56h/57h | 02h | DMA channel 0 element count register |
| DMSFC0 | 56h/57h | 03h | DMA channel 0 sync select and frame count register |
| DMMCR0 | 56h/57h | 04h | DMA channel 0 transfer mode control register |
| DMSRC1 | 56h/57h | 05h | DMA channel 1 source address register |
| DMDST1 | 56h/57h | 06h | DMA channel 1 destination address register |
| DMCTR1 | 56h/57h | 07h | DMA channel 1 element count register |
| DMSFC1 | 56h/57h | 08h | DMA channel 1 sync select and frame count register |
| DMMCR1 | 56h/57h | 09h | DMA channel 1 transfer mode control register |
| DMSRC2 | 56h/57h | 0Ah | DMA channel 2 source address register |
| DMDST2 | 56h/57h | 0Bh | DMA channel 2 destination address register |
| DMCTR2 | 56h/57h | 0Ch | DMA channel 2 element count register |
| DMSFC2 | 56h/57h | 0Dh | DMA channel 2 sync select and frame count register |
| DMMCR2 | 56h/57h | 0Eh | DMA channel 2 transfer mode control register |
| DMSRC3 | 56h/57h | 0Fh | DMA channel 3 source address register |
| DMDST3 | 56h/57h | 10h | DMA channel 3 destination address register |
| DMCTR3 | 56h/57h | 11h | DMA channel 3 element count register |
| DMSFC3 | 56h/57h | 12h | DMA channel 3 sync select and frame count register |
| DMMCR3 | 56h/57h | 13h | DMA channel 3 transfer mode control register |
| DMSRC4 | 56h/57h | 14h | DMA channel 4 source address register |
| DMDST4 | 56h/57h | 15h | DMA channel 4 destination address register |
| DMCTR4 | 56h/57h | 16h | DMA channel 4 element count register |
| DMSFC4 | 56h/57h | 17h | DMA channel 4 sync select and frame count register |
| DMMCR4 | 56h/57h | 18h | DMA channel 4 transfer mode control register |
| DMSRC5 | 56h/57h | 19h | DMA channel 5 source address register |
| DMDST5 | 56h/57h | 1Ah | DMA channel 5 destination address register |
| DMCTR5 | 56h/57h | 1Bh | DMA channel 5 element count register |
| DMSFC5 | 56h/57h | 1Ch | DMA channel 5 sync select and frame count register |
| DMMCR5 | 56h/57h | 1Dh | DMA channel 5 transfer mode control register |
| DMSRCP | 56h/57h | 1Eh | DMA source program page address (common channel) |

**Table 3-18. DMA Subbank Addressed Registers (continued)**

| NAME | ADDRESS | SUB- ADDRESS | DESCRIPTION |
|---|---|---|---|
| DMDSTP | 56h/57h | 1Fh | DMA destination program page address (common channel) |
| DMIDX0 | 56h/57h | 20h | DMA element index address register 0 |
| DMIDX1 | 56h/57h | 21h | DMA element index address register 1 |
| DMFRI0 | 56h/57h | 22h | DMA frame index register 0 |
| DMFRI1 | 56h/57h | 23h | DMA frame index register 1 |
| DMGSA0 | 56h/57h | 24h | DMA global source address reload register, channel 0 |
| DMGDA0 | 56h/57h | 25h | DMA global destination address reload register, channel 0 |
| DMGCR0 | 56h/57h | 26h | DMA global count reload register, channel 0 |
| DMGFR0 | 56h/57h | 27h | DMA global frame count reload register, channel 0 |
| XSRCDP | 56h/57h | 28h | DMA extended source data page (currently not supported) |
| XDSTDP | 56h/57h | 29h | DMA extended destination data page (currently not supported) |
| DMGSA1 | 56h/57h | 2Ah | DMA global source address reload register, channel 1 |
| DMGDA1 | 56h/57h | 2Bh | DMA global destination address reload register, channel 1 |
| DMGCR1 | 56h/57h | 2Ch | DMA global count reload register, channel 1 |
| DMGFR1 | 56h/57h | 2Dh | DMA global frame count reload register, channel 1 |
| DMGSA2 | 56h/57h | 2Eh | DMA global source address reload register, channel 2 |
| DMGDA2 | 56h/57h | 2Fh | DMA global destination address reload register, channel 2 |
| DMGCR2 | 56h/57h | 30h | DMA global count reload register, channel 2 |
| DMGFR2 | 56h/57h | 31h | DMA global frame count reload register, channel 2 |
| DMGSA3 | 56h/57h | 32h | DMA global source address reload register, channel 3 |
| DMGDA3 | 56h/57h | 33h | DMA global destination address reload register, channel 3 |
| DMGCR3 | 56h/57h | 34h | DMA global count reload register, channel 3 |
| DMGFR3 | 56h/57h | 35h | DMA global frame count reload register, channel 3 |
| DMGSA4 | 56h/57h | 36h | DMA global source address reload register, channel 4 |
| DMGDA4 | 56h/57h | 37h | DMA global destination address reload register, channel 4 |
| DMGCR4 | 56h/57h | 38h | DMA global count reload register, channel 4 |
| DMGFR4 | 56h/57h | 39h | DMA global frame count reload register, channel 4 |
| DMGSA5 | 56h/57h | 3Ah | DMA global source address reload register, channel 5 |
| DMGDA5 | 56h/57h | 3Bh | DMA global destination address reload register, channel 5 |
| DMGCR5 | 56h/57h | 3Ch | DMA global count reload register, channel 5 |
| DMGFR5 | 56h/57h | 3Dh | DMA global frame count reload register, channel 5 |

## 3.18 Interrupts

Vector-relative locations and priorities for all internal and external interrupts are shown in Table 3-19.

**Table 3-19. Interrupt Locations and Priorities**

| NAME | TRAP/INTR NUMBER (K) | LOCATION DECIMAL HEX | | PRIORITY | FUNCTION |
|---|---|---|---|---|---|
| $\overline{\text{RS}}$, SINTR | 0 | 0 | 00 | 1 | Reset (hardware and software reset) |
| $\overline{\text{NMI}}$, SINT16 | 1 | 4 | 04 | 2 | Nonmaskable interrupt |
| SINT17 | 2 | 8 | 08 | — | Software interrupt #17 |
| SINT18 | 3 | 12 | 0C | — | Software interrupt #18 |
| SINT19 | 4 | 16 | 10 | — | Software interrupt #19 |
| SINT20 | 5 | 20 | 14 | — | Software interrupt #20 |
| SINT21 | 6 | 24 | 18 | — | Software interrupt #21 |
| SINT22 | 7 | 28 | 1C | — | Software interrupt #22 |
| SINT23 | 8 | 32 | 20 | — | Software interrupt #23 |
| SINT24 | 9 | 36 | 24 | — | Software interrupt #24 |
| SINT25 | 10 | 40 | 28 | — | Software interrupt #25 |
| SINT26 | 11 | 44 | 2C | — | Software interrupt #26 |
| SINT27 | 12 | 48 | 30 | — | Software interrupt #27 |
| SINT28 | 13 | 52 | 34 | — | Software interrupt #28 |
| SINT29 | 14 | 56 | 38 | — | Software interrupt #29 |
| SINT30 | 15 | 60 | 3C | — | Software interrupt #30 |
| $\overline{\text{INT0}}$, SINT0 | 16 | 64 | 40 | 3 | External user interrupt #0 |
| $\overline{\text{INT1}}$, SINT1 | 17 | 68 | 44 | 4 | External user interrupt #1 |
| $\overline{\text{INT2}}$, SINT2 | 18 | 72 | 48 | 5 | External user interrupt #2 |
| TINT, SINT3 | 19 | 76 | 4C | 6 | Timer interrupt |
| RINT0, SINT4 | 20 | 80 | 50 | 7 | McBSP #0 receive interrupt (default) |
| XINT0, SINT5 | 21 | 84 | 54 | 8 | McBSP #0 transmit interrupt (default) |
| RINT2, SINT6 | 22 | 88 | 58 | 9 | McBSP #2 receive interrupt (default) |
| XINT2, SINT7 | 23 | 92 | 5C | 10 | McBSP #2 transmit interrupt (default) |
| $\overline{\text{INT3}}$, SINT8 | 24 | 96 | 60 | 11 | External user interrupt #3 |
| $\overline{\text{HINT}}$, SINT9 | 25 | 100 | 64 | 12 | HPI interrupt |
| RINT1, SINT10 | 26 | 104 | 68 | 13 | McBSP #1 receive interrupt (default) |
| XINT1, SINT11 | 27 | 108 | 6C | 14 | McBSP #1 transmit interrupt (default) |
| DMAC4,SINT12 | 28 | 112 | 70 | 15 | DMA channel 4 (default) |
| DMAC5,SINT13 | 29 | 116 | 74 | 16 | DMA channel 5 (default) |
| Reserved | 30-31 | 120-127 | 78-7F | — | Reserved |

The bit layout of the interrupt flag register (IFR) and the interrupt mask register (IMR) is shown in Figure 3-23.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | DMAC5 | DMAC4 | XINT1 | RINT1 | $\overline{\text{HINT}}$ | $\overline{\text{INT3}}$ |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| XINT2 | RINT2 | XINT0 | RINT0 | TINT | $\overline{\text{INT2}}$ | $\overline{\text{INT1}}$ | $\overline{\text{INT0}}$ |

**LEGEND:** R = Read, W = Write, *n* = value present after reset

**Figure 3-23. IFR and IMR Registers**

# 4 Support

## 4.1 Documentation Support

Extensive documentation supports all TMS320™ DSP family of devices from product announcement through applications development. The following types of documentation are available to support the design and use of the C5000™ platform of DSPs:

**SPRU307:** TMS320C54x DSP Family Functional Overview

Provides a functional overview of the devices included in the TMS320C54xE DSP generation of digital signal processors. Included are descriptions of the CPU architecture, bus structure, memory structure, on-chip peripherals, and instruction set.

**SPRA164:** Calculation of TMS320LC54x Power Dissipation

Describes the power-saving features of the TMS320LC54x and presents techniques for analyzing systems and device conditions to determine operating current levels and power dissipaton. From this information, informed decisions can be made regarding power supply requirements and thermal management considerations.

The five-volume *TMS320C54x DSP Reference Set* consists of:

**SPRU131:** TMS320C54x DSP Reference Set, Volume 1: CPU

Describes the TMS320C54x 16-bit fixed-point general-purpose digital signal processors. Covered are its architecture, internal register structure, data and program addressing, and the instruction pipeline. Also includes development support information, parts lists, and design considerations for using the XDS510 emulator.

**SPRU172:** TMS320C54x DSP Reference Set, Volume 2: Mnemonic Instruction Set

Describes the TMS320C54x digital signal processor mnemonic instructions individually. Also includes a summary of instruction set classes and cycles.

**SPRU179:** TMS320C54x DSP Reference Set, Volume 3: Algebraic Instruction Set

Describes the TMS320C54x digital signal processor algebraic instructions individually. Also includes a summary of instruction set classes and cycles.

**SPRU173:** TMS320C54x DSP Reference Set, Volume 4: Applications Guide

Describes software and hardware applications for the TMS320C54x digital signal processor. Also includes development support information, parts lists, and design considerations for using the XDS510 emulator.

**SPRU302:** TMS320C54x DSP Reference Set, Volume 5: Enhanced Peripherals

Describes the enhanced peripherals available on the TMS320C54x digital signal processors. Includes the multichannel buffered serial ports (McBSPs), direct memory access (DMA) controller, interprocessor communications, and the HPI-8 and HPI-16 host port interfaces.

The reference set describes in detail the TMS320C54x™ DSP products currently available and the hardware and software applications, including algorithms, for fixed-point TMS320™ DSP family of devices.

A series of DSP textbooks is published by Prentice-Hall and John Wiley & Sons to support digital signal processing research and education. The TMS320 DSP newsletter, *Details on Signal Processing*, is published quarterly and distributed to update TMS320 DSP customers on product information.

Information regarding TI DSP porducts is also available on the web at *www.ti.com*.

## 4.2 Device and Development-Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all TMS320 DSP devices and support tools. Each TMS320 DSP commercial family member has one of three prefixes: TMX, TMP, or TMS (e.g., TMS320C6412GDK600). Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

Device development evolutionary flow:

**TMX** Experimental device that is not necessarily representative of the final device's electrical specifications

**TMP** Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification

**TMS** Fully qualified production device

Support tool development evolutionary flow:

**TMDX** Development-support product that has not yet completed Texas Instruments internal qualification testing.

**TMDS** Fully qualified development-support product

TMX and TMP devices and TMDX development-support tools are shipped with appropriate disclaimers describing their limitations and intended uses.

"Developmental product is intended for internal evaluation purposes."

TMS devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that prototype devices (TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

# 5 Electrical Specifications

This section provides the absolute maximum ratings and the recommended operating conditions for the TMS320VC5416 DSP.

## 5.1 Absolute Maximum Ratings

The absolute maximum ratings are measure over operating case temperature range. Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under Section Section 5.3.1 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability. All voltage values are with respect to $DV_{SS}$.

| | | |
|---|---|---|
| $DV_{DD}$ | Supply voltage I/O range | – 0.3 V to 4.0 V |
| $CV_{DD}$ | Supply voltage core range | – 0.3 V to 2.0 V |
| $V_I$ | Input voltage range | – 0.3 V to 4.5 V |
| $V_O$ | Output voltage range | – 0.3 V to 4.5 V |
| $T_C$ | Operating case temperature range | – 40°C to 100°C |
| $T_{stg}$ | Storage temperature range | – 55°C to 150°C |

## 5.2 Recommended Operating Conditions

| | | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| $DV_{DD}$ | Device supply voltage, I/O | | 2.7 | 3.3 | 3.6 | V |
| $CV_{DD}$ | Device supply voltage, core (VC5416-160) | | 1.55 | 1.6 | 1.65 | V |
| $CV_{DD}$ | Device supply voltage, core (VC5416-120) | | 1.42 | 1.5 | 1.65 | V |
| $DV_{SS}$<br>$CV_{SS}$ | Supply voltage, GND | | | 0 | | V |
| $V_{IH}$ | High-level input voltage, I/O | $\overline{\text{RS}}$, $\overline{\text{INTn}}$, $\overline{\text{NMI}}$, X2/CLKIN, CLKMDn, BCLKRn, BCLKXn, $\overline{\text{HCS}}$, $\overline{\text{HDS1}}$, $\overline{\text{HDS2}}$, $\overline{\text{HAS}}$, $\overline{\text{TRST}}$, $\overline{\text{BIO}}$, Dn, An, HDn, TCK<br>$DV_{DD}$ = 2.7 V to 3.6 V | 2.4 | | $DV_{DD}$ + 0.3 | V |
| | | All other inputs | 2 | | $DV_{DD}$ + 0.3 | |
| $V_{IL}$ | Low-level input voltage | | – 0.3 | | 0.8 | V |
| $I_{OH}$ | High-level output current(1)(2) | | | | – 8 | mA |
| $I_{OL}$ | Low-level output current(1)(2) | | | | 8 | mA |
| $T_C$ | Operating case temperature | | – 40 | | 100 | °C |

(1) The maximum output currents are DC values only. Transient currents may exceed these values.
(2) These output current limits are used for the test conditions on $V_{OL}$ and $V_{OH}$, except where noted otherwise.

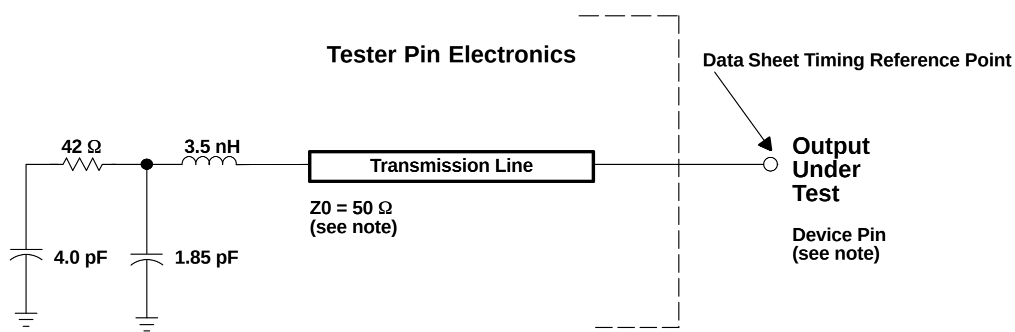
## 5.3 Electrical Characteristics

The electrical charactheristics are measured over recommended operating case temperature range (unless otherwise noted). All values are typical unless otherwise specified.

| PARAMETER | | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $V_{OH}$ | High-level output voltage (1) | | $DV_{DD}$ = 2.7 V to 3 V, $I_{OH}$ = -2 mA | 2.2 | | | V |
| | | | $DV_{DD}$ = 3 V to 3.6 V, $I_{OH}$ = MAX | 2.4 | | | |
| $V_{OL}$ | Low-level output voltage (1) | | $I_{OL}$ = MAX | | | 0.4 | V |
| $I_I$ | Input current ($V_I$ = $DV_{SS}$ to $DV_{DD}$) | X2/CLKIN | | – 40 | | 40 | μA |
| | | $\overline{TRST}$, HPI16 | With internal pulldown | – 10 | | 800 | μA |
| | | HPIENA | With internal pulldown, $\overline{RS}$ = 0 | – 10 | | 400 | |
| | | TMS, TCK, TDI, HPI (2) | With internal pullups | – 400 | | 10 | |
| | | A[17:0], D[15:0], HD[7:0] | Bus holders enabled, $DV_{DD}$ = MAX (3) | – 275 | | 275 | |
| | | All other input-only pins | | – 5 | | 5 | |
| $I_{DDC}$ | Supply current, core CPU | | $CV_{DD}$ = 1.6 V, $f_x$ = 160 , (4) $T_C$ = 25°C | | 60 (5) | | mA |
| $I_{DDP}$ | Supply current, pins | | $DV_{DD}$ = 3.0 V, $f_x$ = 160 MHz, (4) $T_C$ = 25°C | | 40 (6) | | mA |
| $I_{DD}$ | Supply current, standby | IDLE2 | PLL × 1 mode, 20 MHz input | | 2 | | mA |
| | | IDLE3 Divide-by-two mode, CLKIN stopped | $T_C$ = 25°C | | 1 | | |
| | | | $T_C$ = 100°C | | 30 | | |
| $C_i$ | Input capacitance | | | | 5 | | pF |
| $C_o$ | Output capacitance | | | | 5 | | pF |

(1) All input and output voltage levels except $\overline{RS}$, $\overline{INT0}$-$\overline{INT3}$, $\overline{NMI}$, X2/CLKIN, CLKMD1-CLKMD3, BCLKRn, BCLKXn, $\overline{HCS}$, $\overline{HAS}$, $\overline{HDS1}$, $\overline{HDS2}$, $\overline{BIO}$, TCK, $\overline{TRST}$, Dn, An, HDn are LVTTL-compatible.
(2) HPI input signals except for HPIENA and HPI16, when HPIENA = 0.
(3) $V_{IL(MIN)} \le V_I \le V_{IL(MAX)}$ or $V_{IH(MIN)} \le V_I \le V_{IH(MAX)}$
(4) Clock mode: PLL × 1 with external source
(5) This value was obtained with 50% usage of MAC and 50% usage of NOP instructions. Actual operating current varies with program being executed.
(6) This value was obtained with single-cycle external writes, CLKOFF = 0 and load = 15 pF. For more details on how this calculation is performed, refer to the *Calculation of TMS320LC54x Power Dissipation* application report (literature number SPRA164).

### *5.3.1 Test Loading*




NOTE: The data sheet provides timing at the device pin. For output timing analysis, the tester pin electronics and its transmission line effects must taken into account. A transmission line with a delay of 2 ns or longer can be used to produce the desired transmission line effect. The transmission line is intended as a load only. It is not necessary to add or subtract the transmission line delay (2 ns or longer) from the data sheet timings.

Input requirements in this data sheet are tested with an input slew rate of < 4 Volts per nanosecond (4 V/ns) at the device pin.

**Figure 5-1. Tester Pin Electronics**

### 5.3.2 Timing Parameter Symbology

Timing parameter symbols used in the timing requirements and switching characteristics tables are created in accordance with JEDEC Standard 100. To shorten the symbols, some of the pin names and other related terminology have been abbreviated as follows:

**Lowercase subscripts and their meanings:**

| | |
|---|---|
| a | access time |
| c | cycle time (period) |
| d | delay time |
| dis | disable time |
| en | enable time |
| f | fall time |
| h | hold time |
| r | rise time |
| su | setup time |
| t | transition time |
| v | valid time |
| w | pulse duration (width) |
| X | Unknown, changing, or don't care level |

**Letters and symbols and their meanings:**

| | |
|---|---|
| H | High |
| L | Low |
| V | Valid |
| Z | High impedance |

### 5.3.3 Internal Oscillator With External Crystal

The internal oscillator is enabled by selecting the appropriate clock mode at reset (this is device-dependent; see Section Section 3.10) and connecting a crystal or ceramic resonator across X1 and X2/CLKIN. The CPU clock frequency is one-half, one-fourth, or a multiple of the oscillator frequency. The multiply ratio is determined by the bit settings in the CLKMD register.

The crystal should be in fundamental-mode operation, and parallel resonant, with an effective series resistance of 30 Ω maximum and power dissipation of 1 mW. The connection of the required circuit, consisting of the crystal and two load capacitors, is shown in Figure 5-2. The load capacitors, $C_1$ and $C_2$, should be chosen such that the equation below is satisfied. $C_L$ (recommended value of 10 pF) in the equation is the load specified for the crystal.

$$C_L = \frac{C_1 C_2}{(C_1 + C_2)}$$

**Table 5-1. Input Clock Frequency Characteristics**

| | | MIN | MAX | Unit |
|---|---|---|---|---|
| $f_x$ | Input clock frequency | 10[(1)] | 20[(2)] | MHz |

(1) This device utilizes a fully static design and therefore can operate with $t_{c(CI)}$ approaching ∞. The device is characterized at frequencies approaching 0 Hz
(2) It is recommended that the PLL multiply by N clocking option be used for maximum frequency operation.

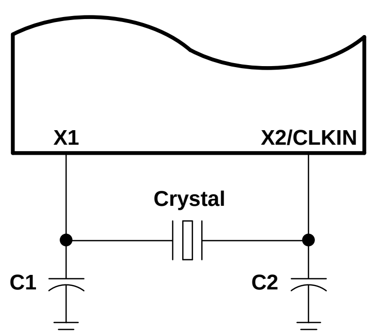



**Figure 5-2. Internal Divide-By-Two Clock Option With External Crystal**

## 5.4 Clock Options

The frequency of the reference clock provided at the CLKIN pin can be divided by a factor of two or four or multiplied by one of several values to generate the internal machine cycle.

### 5.4.1 Divide-By-Two and Divide-By-Four Clock Options

The frequency of the reference clock provided at the X2/CLKIN pin can be divided by a factor of two or four to generate the internal machine cycle. The selection of the clock mode is described in Section Section 3.10.

When an external clock source is used, the frequency injected must conform to specifications listed in Table 5-3.

An external frequency source can be used by applying an input clock to X2/CLKIN with X1 left unconnected.

Table 5-2 shows the configuration options for the CLKMD pins that generate the external divide-by-2 or divide-by-4 clock option. Table 5-3 and Table 5-4 assume testing over recommended operating conditions and H = $0.5t_{c(CO)}$ (see Figure 5-3).

**Table 5-2. Clock Mode Pin Settings for the Divide-By-2 and By Divide-By-4 Clock Options**

| CLKMD1 | CLKMD2 | CLKMD3 | Clock Mode |
|---|---|---|---|
| 0 | 0 | 0 | 1/2, PLL disabled |
| 1 | 0 | 1 | 1/4, PLL disabled |
| 1 | 1 | 1 | 1/2, PLL disabled |

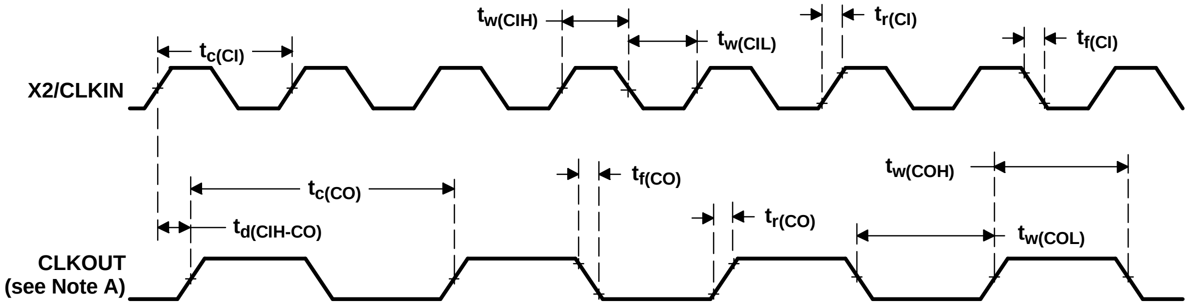
**Table 5-3. Divide-By-2 and Divide-By-4 Clock Options Timing Requirements**

| | | 5416-120 5416-160 | | Unit |
|---|---|---|---|---|
| | | MIN | MAX | |
| $t_{c(CI)}$ | Cycle time, X2/CLKIN | 20 | | ns |
| $t_{f(CI)}$ | Fall time, X2/CLKIN | | 4 | ns |
| $t_{r(CI)}$ | Rise time, X2/CLKIN | | 4 | ns |
| $t_{w(CIL)}$ | Pulse duration, X2/CLKIN low | 4 | | ns |
| $t_{w(CIH)}$ | Pulse duration, X2/CLKIN high | 4 | | ns |

**Table 5-4. Divide-By-2 and Divide-By-4 Clock Options Switching Characteristics**

| PARAMETER | | 5416-120 | | | 5416-160 | | | Unit |
|---|---|---|---|---|---|---|---|---|
| | | MIN | TYP | MAX | MIN | TYP | MAX | |
| $t_{c(CO)}$ | Cycle time, CLKOUT | 8.33(1) | | (2) | 6.25(1) | | (2) | ns |
| $t_{d(CIH-CO)}$ | Delay time, X2/CLKIN high to CLKOUT high/low | 4 | 7 | 11 | 4 | 7 | 11 | ns |
| $t_{f(CO)}$ | Fall time, CLKOUT | | 1 | | | 1 | | ns |
| $t_{r(CO)}$ | Rise time, CLKOUT | | 1 | | | 1 | | ns |
| $t_{w(COL)}$ | Pulse duration, CLKOUT low | H – 2 | H | H + 1 | H – 2 | H | H + 1 | ns |
| $t_{w(COH)}$ | Pulse duration, CLKOUT high | H – 2 | H | H + 1 | H – 2 | H | H + 1 | ns |

(1) It is recommended that the PLL clocking operation be used for maximum frequency operation.
(2) This device utilizes a fully static design and therefore can operate with $t_{c(CI)}$ approaching ∞. The device is characterized at frequencies approaching 0 Hz.




A. The CLKOUT timing in this diagram assumes the CLKOUT divide factor (DIVFCT field in the BSCR) is configured as 00 (CLKOUT not divided). DIVFCT is configured as CLKOUT divided-by-4 mode following reset.

**Figure 5-3. External Divide-By-Two Clock Timing**

### 5.4.2 Multiply-By-N Clock Option (PLL Enabled)

The frequency of the reference clock provided at the X2/CLKIN pin can be multiplied by a factor of N to generate the internal machine cycle. The selection of the clock mode and the value of N is described in Section Section 3.10. Following reset, the software PLL can be programmed for the desired multiplication factor. Refer to the *TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals* (literature number SPRU131) for detailed information on programming the PLL.

When an external clock source is used, the external frequency injected must conform to specifications listed in Table 5-5.

Table 5-5 and Table 5-6 assume testing over recommended operating conditions and H = $0.5t_{c(CO)}$ (see Figure 5-4).

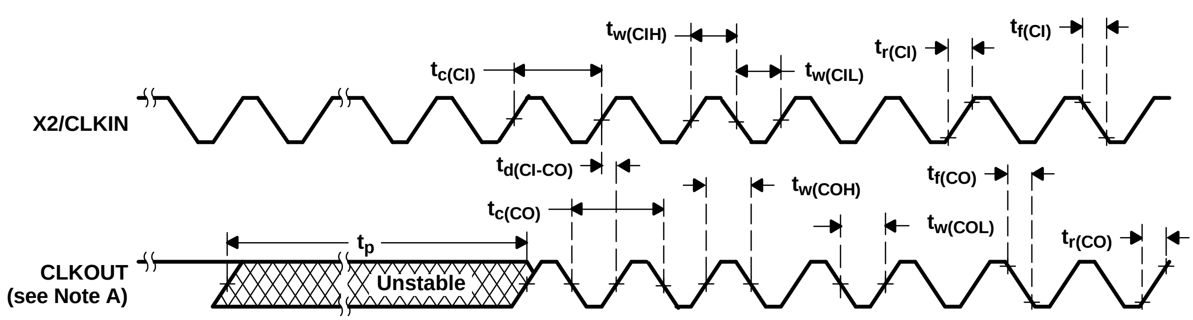
**Table 5-5. Multiply-By-N Clock Option Timing Requirements**

| | | | 5416-120 5416-160 MIN | MAX | Unit |
|---|---|---|---|---|---|
| $t_{c(CI)}$ | Cycle time, X2/CLKIN | Integer PLL multiplier N (N = 1-15)(1) | 20 | 200 | ns |
| | | PLL multiplier N = x.5(1) | 20 | 100 | |
| | | PLL multiplier N = x.25, x.75(1) | 20 | 50 | |
| $t_{f(CI)}$ | Fall time, X2/CLKIN | | | 4 | ns |
| $t_{r(CI)}$ | Rise time, X2/CLKIN | | | 4 | ns |
| $t_{w(CIL)}$ | Pulse duration, X2/CLKIN low | | 4 | | ns |
| $t_{w(CIH)}$ | Pulse duration, X2/CLKIN high | | 4 | | ns |

(1) N is the multiplication factor.

**Table 5-6. Multiply-By-N Clock Option Switching Characteristics**

| PARAMETER | | 5416-120 MIN | TYP | MAX | 5416-160 MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|---|---|---|
| $t_{c(CO)}$ | Cycle time, CLKOUT | 8.33 | | | 6.25 | | | ns |
| $t_{d(CI-CO)}$ | Delay time, X2/CLKIN high/low to CLKOUT high/low | 4 | 7 | 11 | 4 | 7 | 11 | ns |
| $t_{f(CO)}$ | Fall time, CLKOUT | | 2 | | | 2 | | ns |
| $t_{r(CO)}$ | Rise time, CLKOUT | | 2 | | | 2 | | ns |
| $t_{w(COL)}$ | Pulse duration, CLKOUT low | H – 2 | H | H + 1 | H – 2 | H | H + 1 | ns |
| $t_{w(COH)}$ | Pulse duration, CLKOUT high | H – 2 | H | H + 1 | H – 2 | H | H + 1 | ns |
| $t_p$ | Transitory phase, PLL lock-up time | | | 30 | | | 30 | ms |




A. The CLKOUT timing in this diagram assumes the CLKOUT divide factor (DIVFCT field in the BSCR) is configured as 00 (CLKOUT not divided). DIVFCT is configured as CLKOUT divided-by-4 mode following reset.

**Figure 5-4. Multiply-By-One Clock Timing**

## 5.5 Memory and Parallel I/O Interface Timing

### 5.5.1 Memory Read

External memory reads can be performed in consecutive or nonconsecutive mode under control of the $\overline{\text{CONSEC}}$ bit in the BSCR. Table 5-7 and Table 5-8 assume testing over recommended operating conditions with $\overline{\text{MSTRB}}$ = 0 and H = $0.5t_{c(CO)}$ (see Figure 5-5 and Figure 5-6).

**Table 5-7. Memory Read Timing Requirements**

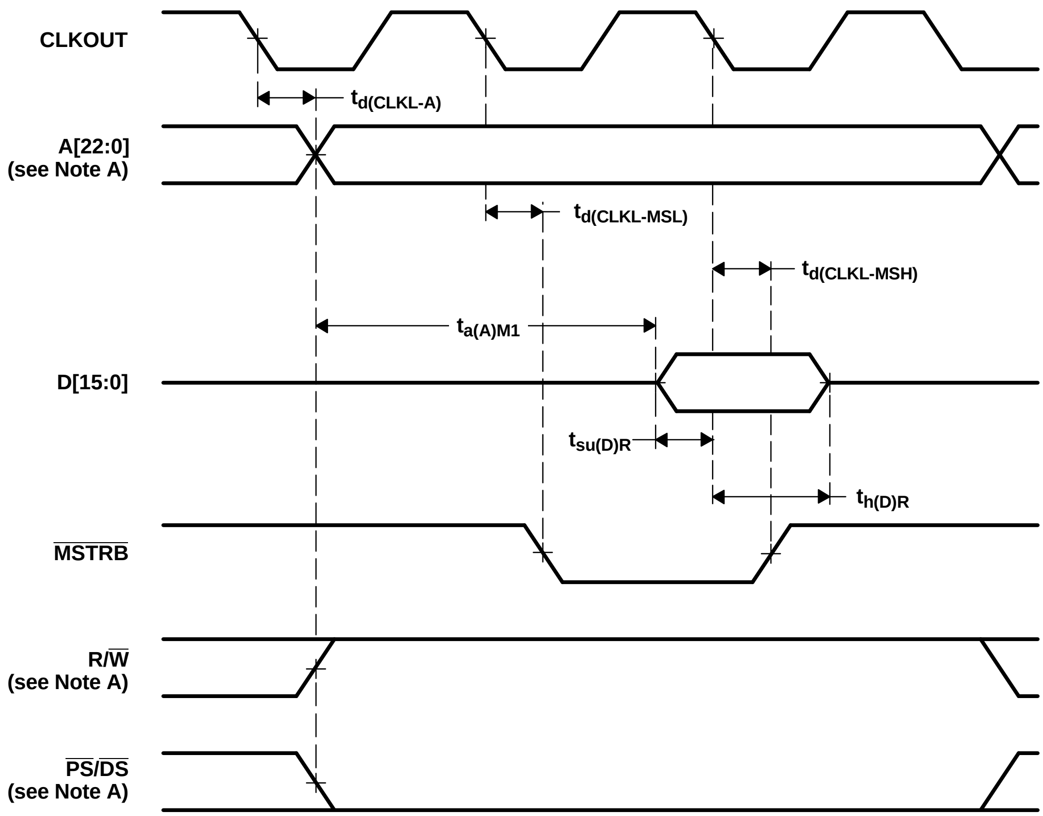
| | | 5416-120 5416-160 | | UNIT |
|---|---|---|---|---|
| | | MIN | MAX | |
| $t_{a(A)M1}$ | Access time, read data access from address valid, first read access(1) | | 4H–9 | ns |
| $t_{a(A)M2}$ | Access time, read data access from address valid, consecutive read accesses(1) | | 2H–9 | ns |
| $t_{su(D)R}$ | Setup time, read data valid before CLKOUT low | 7 | | ns |
| $t_{h(D)R}$ | Hold time, read data valid after CLKOUT low | 0 | | ns |

(1) Address, R/$\overline{\text{W}}$, $\overline{\text{PS}}$, $\overline{\text{DS}}$, and $\overline{\text{IS}}$ timings are all included in timings referenced as address.

**Table 5-8. Memory Read Switching Characteristics**

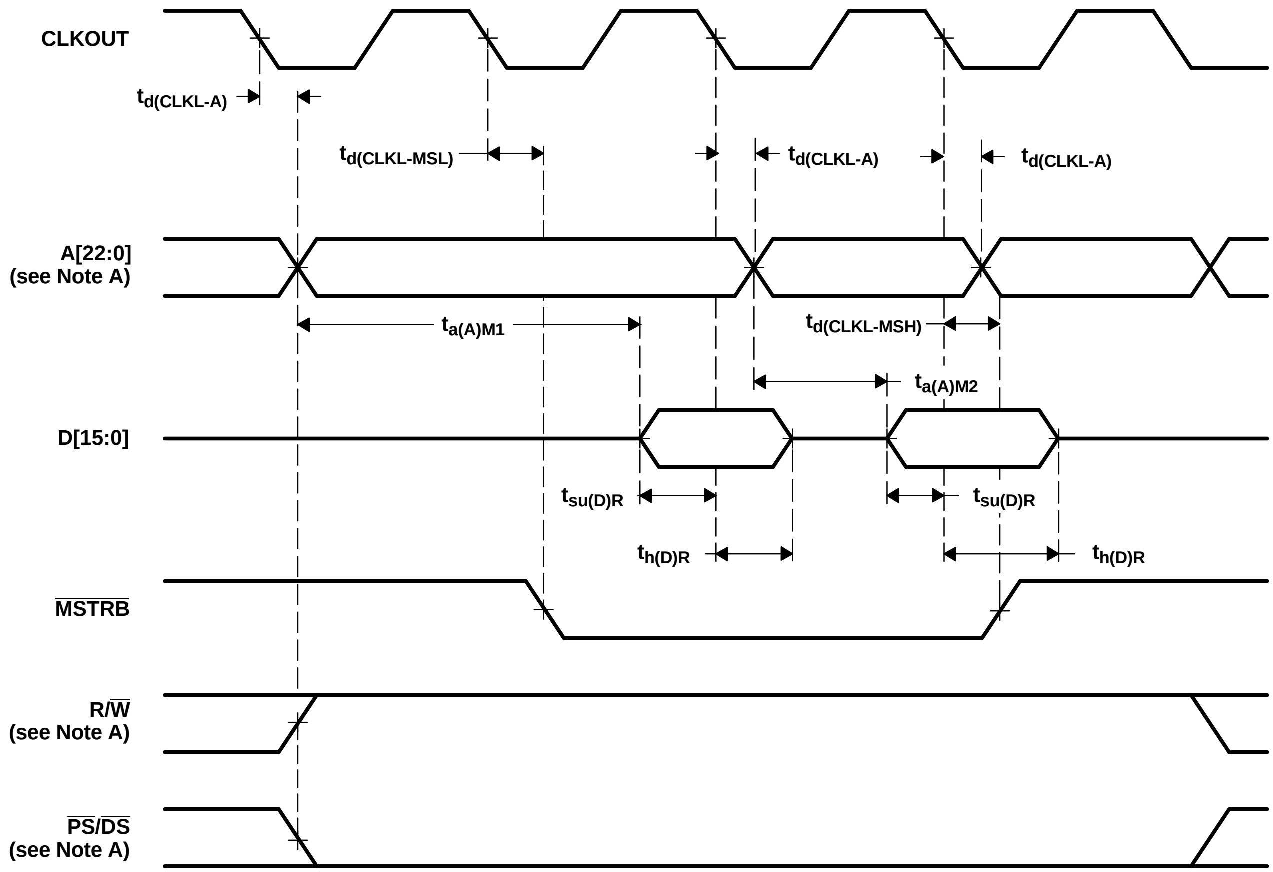
| PARAMETER | | 5416-120 5416-160 | | UNIT |
|---|---|---|---|---|
| | | MIN | MAX | |
| $t_{d(CLKL-A)}$ | Delay time, CLKOUT low to address valid(1) | –1 | 4 | ns |
| $t_{d(CLKL-MSL)}$ | Delay time, CLKOUT low to $\overline{\text{MSTRB}}$ low | –1 | 4 | ns |
| $t_{d(CLKL-MSH)}$ | Delay time, CLKOUT low to $\overline{\text{MSTRB}}$ high | 0 | 4 | ns |

(1) Address, R/$\overline{\text{W}}$, $\overline{\text{PS}}$, $\overline{\text{DS}}$, and $\overline{\text{IS}}$ timings are all included in timings referenced as address.




A. Address, R/$\overline{W}$, $\overline{PS}$, $\overline{DS}$, and $\overline{IS}$ timings are all included in timings referenced as address.

**Figure 5-5. Nonconsecutive Mode Memory Reads**




A. Address, R/W, PS, DS, and IS timings are all included in timings referenced as address.

**Figure 5-6. Consecutive Mode Memory Reads**

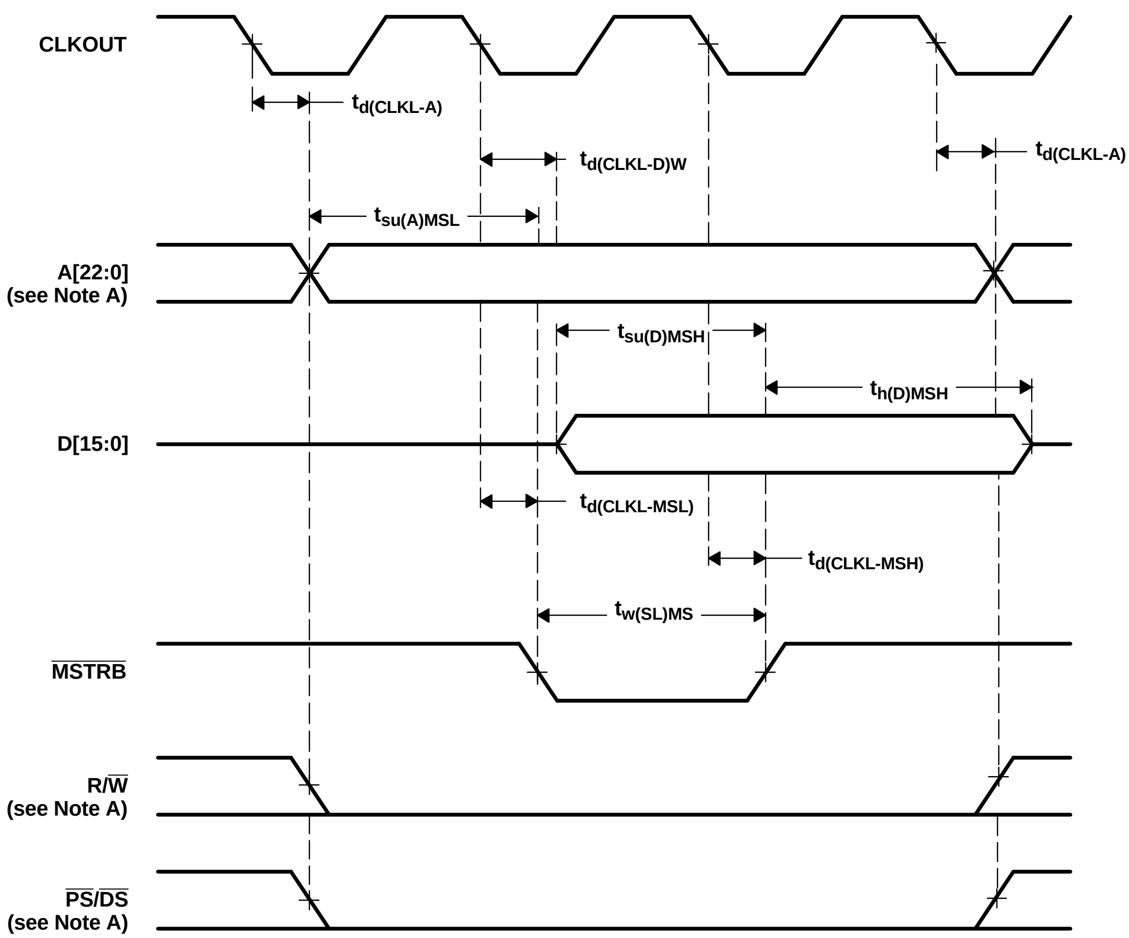
### *5.5.2 Memory Write*

Table 5-9 assumes testing over recommended operating conditions with $\overline{\text{MSTRB}}$ = 0 and H = 0.5$t_{c(CO)}$ (see Figure 5-7).

**Table 5-9. Memory Write Switching Characteristics**

| PARAMETER | | 5416-120 5416-160 MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{d(CLKL-A)}$ | Delay time, CLKOUT low to address valid[1] | –1 | 4 | ns |
| $t_{su(A)MSL}$ | Setup time, address valid before $\overline{\text{MSTRB}}$ low[1] | 2H – 3 | | ns |
| $t_{d(CLKL-D)W}$ | Delay time, CLKOUT low to data valid | –1 | 4 | ns |
| $t_{su(D)MSH}$ | Setup time, data valid before $\overline{\text{MSTRB}}$ high | 2H – 4 | 2H + 6 | ns |
| $t_{h(D)MSH}$ | Hold time, data valid after $\overline{\text{MSTRB}}$ high | 2H – 5 | 2H + 6 | ns |
| $t_{d(CLKL-MSL)}$ | Delay time, CLKOUT low to $\overline{\text{MSTRB}}$ low | –1 | 4 | ns |
| $t_{w(SL)MS}$ | Pulse duration, $\overline{\text{MSTRB}}$ low | 2H – 2 | | ns |
| $t_{d(CLKL-MSH)}$ | Delay time, CLKOUT low to $\overline{\text{MSTRB}}$ high | 0 | 4 | ns |

(1) Address, R/$\overline{\text{W}}$, $\overline{\text{PS}}$, $\overline{\text{DS}}$, and $\overline{\text{IS}}$ timings are all included in timings referenced as address.




A. Address, R/$\overline{\text{W}}$, $\overline{\text{PS}}$, $\overline{\text{DS}}$, and $\overline{\text{IS}}$ timings are all included in timings referenced as address.

**Figure 5-7. Memory Write ($\overline{\text{MSTRB}}$ = 0)**

### 5.5.3 I/O Read

Table 5-10 and Table 5-11 assume testing over recommended operating conditions, $\overline{\text{IOSTRB}}$ = 0, and H = $0.5t_{c(CO)}$ (see Figure 5-8).

**Table 5-10. I/O Read Timing Requirements**

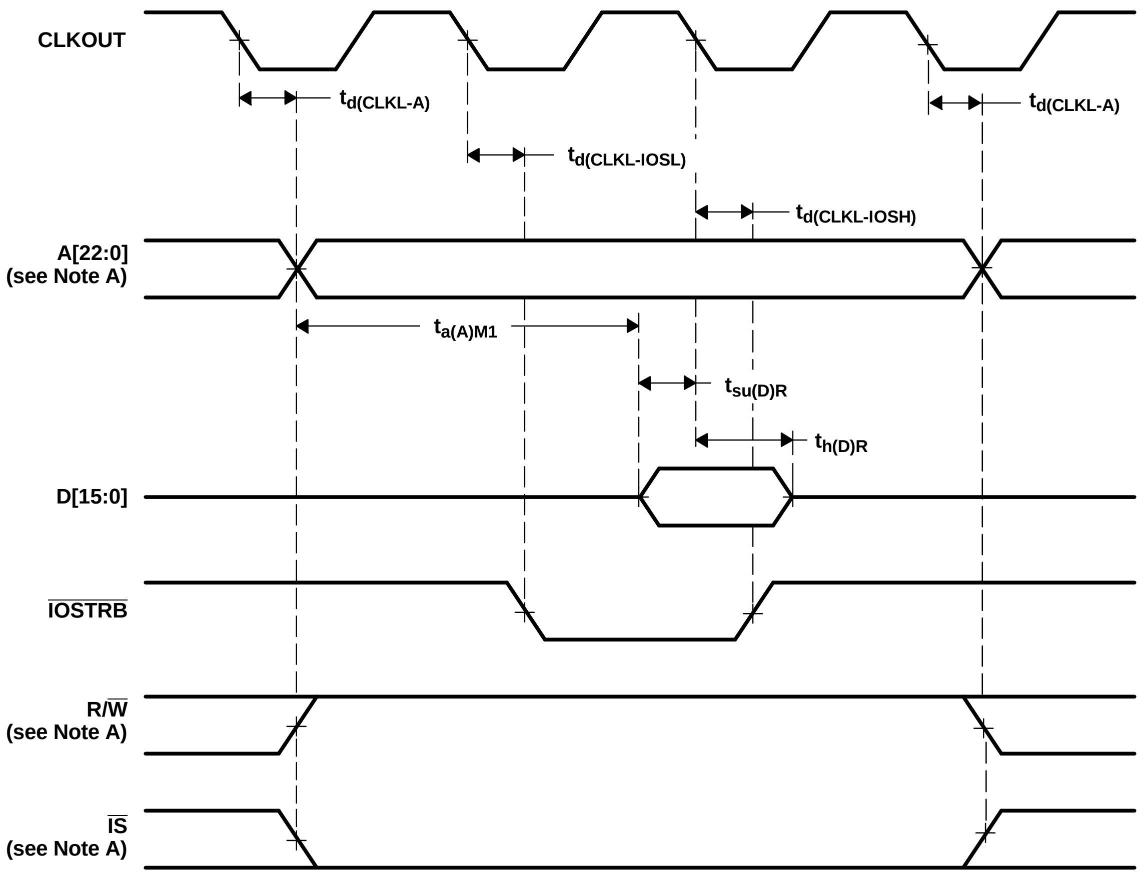
| | | 5416-120<br>5416-160 MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{a(A)M1}$ | Access time, read data access from address valid, first read access[(1)] | | 4H – 9 | ns |
| $t_{su(D)R}$ | Setup time, read data valid before CLKOUT low | 7 | | ns |
| $t_{h(D)R}$ | Hold time, read data valid after CLKOUT low | 0 | | ns |

(1) Address, R/$\overline{\text{W}}$, $\overline{\text{PS}}$, $\overline{\text{DS}}$, and $\overline{\text{IS}}$ timings are all included in timings referenced as address.

**Table 5-11. I/O Read Switching Characteristics**

| PARAMETER | | 5416-120<br>5416-160 MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{d(CLKL-A)}$ | Delay time, CLKOUT low to address valid[(1)] | – 1 | 4 | ns |
| $t_{d(CLKL-IOSL)}$ | Delay time, CLKOUT low to $\overline{\text{IOSTRB}}$ low | – 1 | 4 | ns |
| $t_{d(CLKL-IOSH)}$ | Delay time, CLKOUT low to $\overline{\text{IOSTRB}}$ high | 0 | 4 | ns |

(1) Address, R/$\overline{\text{W}}$, $\overline{\text{PS}}$, $\overline{\text{DS}}$, and $\overline{\text{IS}}$ timings are all included in timings referenced as address.




A. Address, R/$\overline{W}$, $\overline{PS}$, $\overline{DS}$, and $\overline{IS}$ timings are all included in timings referenced as address.

**Figure 5-8. Parallel I/O Port Read ($\overline{IOSTRB}$ = 0)**

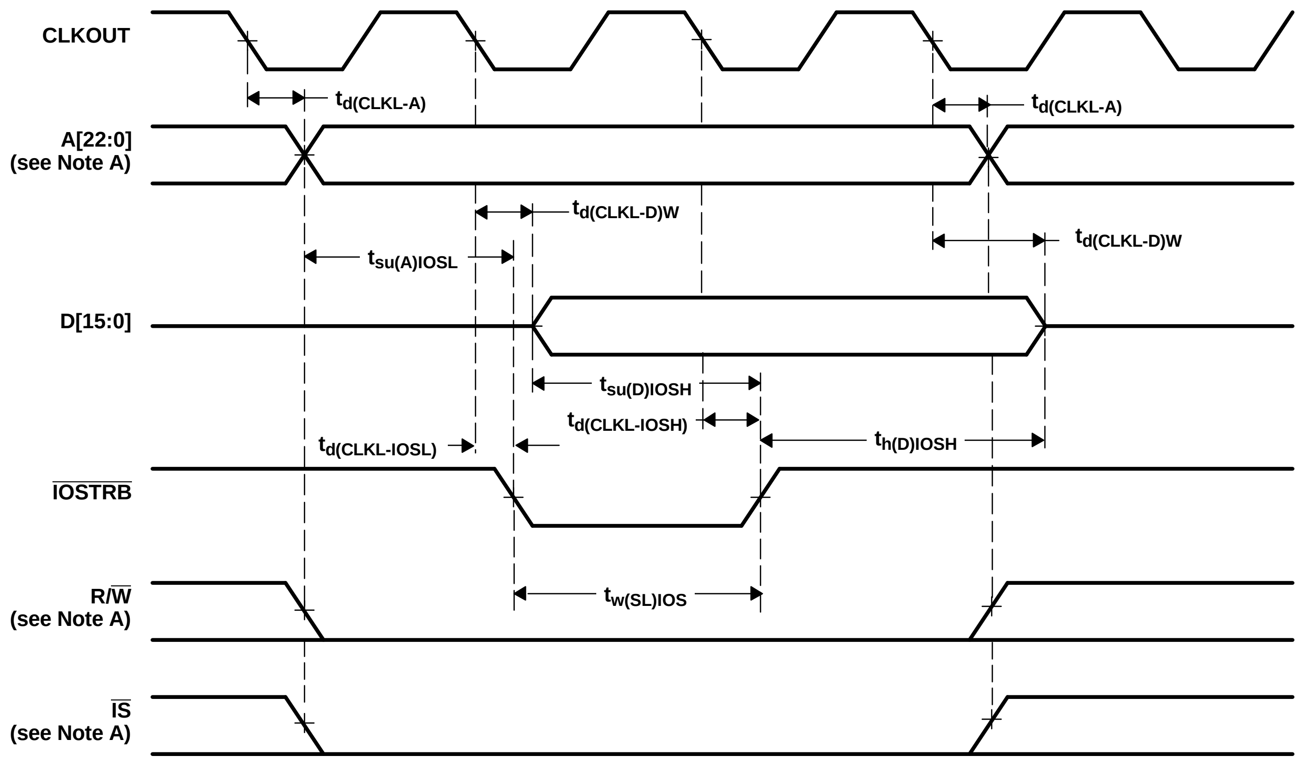
### *5.5.4 I/O Write*

Table 5-12 assumes testing over recommended operating conditions, $\overline{IOSTRB}$ = 0, and H = $0.5t_{c(CO)}$ (see Figure 5-9).

**Table 5-12. I/O Write Switching Characteristics**

| PARAMETER | | 5416-120 5416-160 | | UNIT |
|---|---|---|---|---|
| | | MIN | MAX | |
| $t_{d(CLKL-A)}$ | Delay time, CLKOUT low to address valid(1) | – 1 | 4 | ns |
| $t_{su(A)IOSL}$ | Setup time, address valid before $\overline{IOSTRB}$ low(1) | 2H – 3 | | ns |
| $t_{d(CLKL-D)W}$ | Delay time, CLKOUT low to write data valid | – 1 | 4 | ns |
| $t_{su(D)IOSH}$ | Setup time, data valid before $\overline{IOSTRB}$ high | 2H – 4 | 2H + 6 | ns |
| $t_{h(D)IOSH}$ | Hold time, data valid after $\overline{IOSTRB}$ high | 2H – 5 | 2H + 6 | ns |
| $t_{d(CLKL-IOSL)}$ | Delay time, CLKOUT low to $\overline{IOSTRB}$ low | – 1 | 4 | ns |
| $t_{w(SL)IOS}$ | Pulse duration, $\overline{IOSTRB}$ low | 2H – 2 | | ns |
| $t_{d(CLKL-IOSH)}$ | Delay time, CLKOUT low to $\overline{IOSTRB}$ high | 0 | 4 | ns |

(1) Address, R/$\overline{W}$, $\overline{PS}$, $\overline{DS}$, and $\overline{IS}$ timings are all included in timings referenced as address.




A. Address, R/W̅, P̅S̅, D̅S̅, and I̅S̅ timings are all included in timings referenced as address.

**Figure 5-9. Parallel I/O Port Write (I̅O̅S̅T̅R̅B̅ = 0)**

### 5.5.5 Ready Timing for Externally Generated Wait States

Table 5-13 and Table 5-14 assume testing over recommended operating conditions and H = $0.5t_{c(CO)}$ (see Figure 5-10, Figure 5-11, Figure 5-12, and Figure 5-13).

**Table 5-13. Ready Timing Requirements for Externally Generated Wait States(1)**

| | | 5416-120 5416-160 MIN | 5416-120 5416-160 MAX | UNIT |
|---|---|---|---|---|
| $t_{su(RDY)}$ | Setup time, READY before CLKOUT low | 7 | | ns |
| $t_{h(RDY)}$ | Hold time, READY after CLKOUT low | 0 | | ns |
| $t_{v(RDY)MSTRB}$ | Valid time, READY after $\overline{\text{MSTRB}}$ low(2) | | 4H – 4 | ns |
| $t_{h(RDY)MSTRB}$ | Hold time, READY after $\overline{\text{MSTRB}}$ low(2) | 4H | | ns |
| $t_{v(RDY)IOSTRB}$ | Valid time, READY after $\overline{\text{IOSTRB}}$ low(2) | | 4H – 4 | ns |
| $t_{h(RDY)IOSTRB}$ | Hold time, READY after $\overline{\text{IOSTRB}}$ low(2) | 4H | | ns |

(1) The hardware wait states can be used only in conjunction with the software wait states to extend the bus cycles. To generate wait states by READY, as least two software wait states must be programmed. READY is not sampled until the completion of the internal software wait states.
(2) These timings are included for reference only. The critical timings for READY are those referenced to CLKOUT.

**Table 5-14. Ready Switching Characteristics for Externally Generated Wait States(1)**

| PARAMETER | | 5416-120 5416-160 MIN | 5416-120 5416-160 MAX | UNIT |
|---|---|---|---|---|
| $t_{d(MSCL)}$ | Delay time, CLKOUT low to $\overline{\text{MSC}}$ low | 0 | 4 | ns |
| $t_{d(MSCH)}$ | Delay time, CLKOUT low to $\overline{\text{MSC}}$ high | 0 | 4 | ns |

(1) The hardware wait states can be used only in conjunction with the software wait states to extend the bus cycles. To generate wait states by READY, as least two software wait states must be programmed. READY is not sampled until the completion of the internal software wait states.

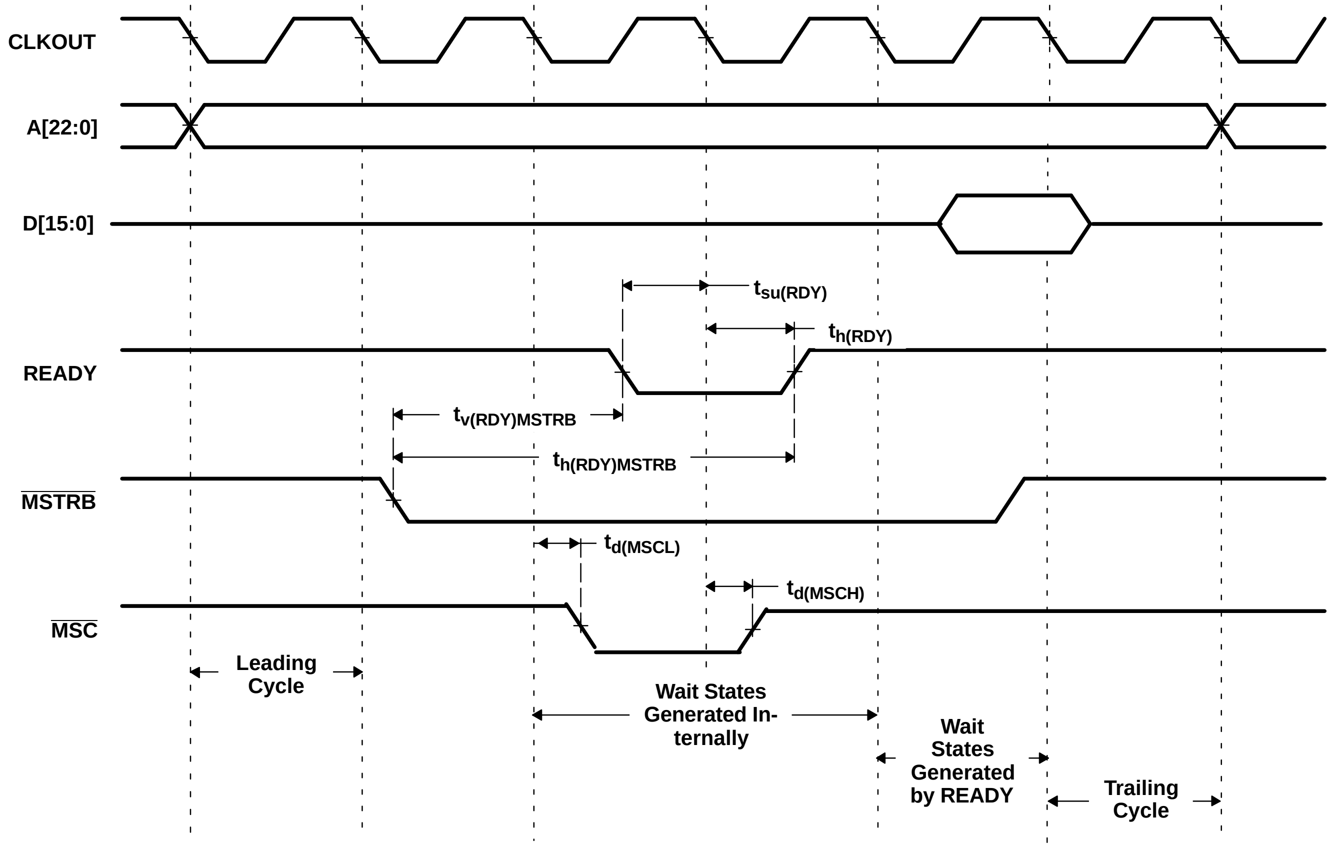



**Figure 5-10. Memory Read With Externally Generated Wait States**

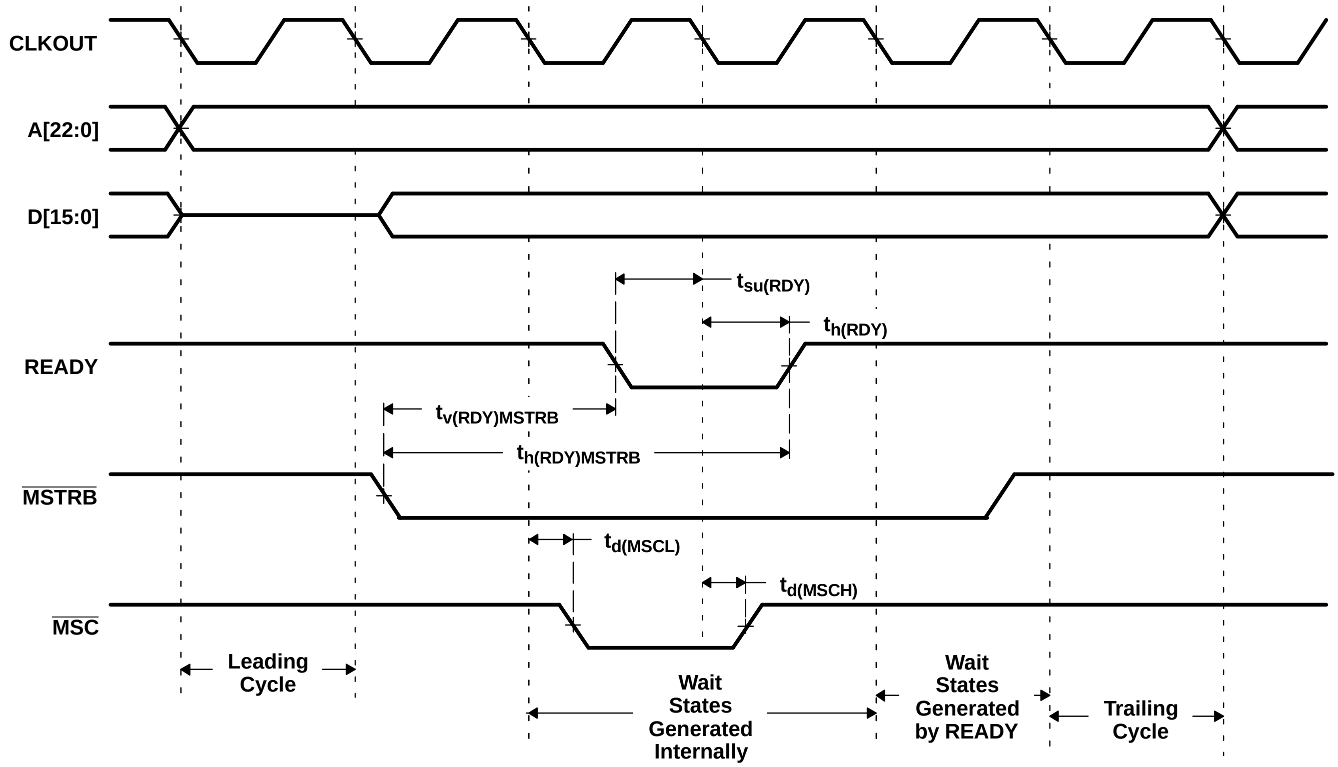



**Figure 5-11. Memory Write With Externally Generated Wait States**

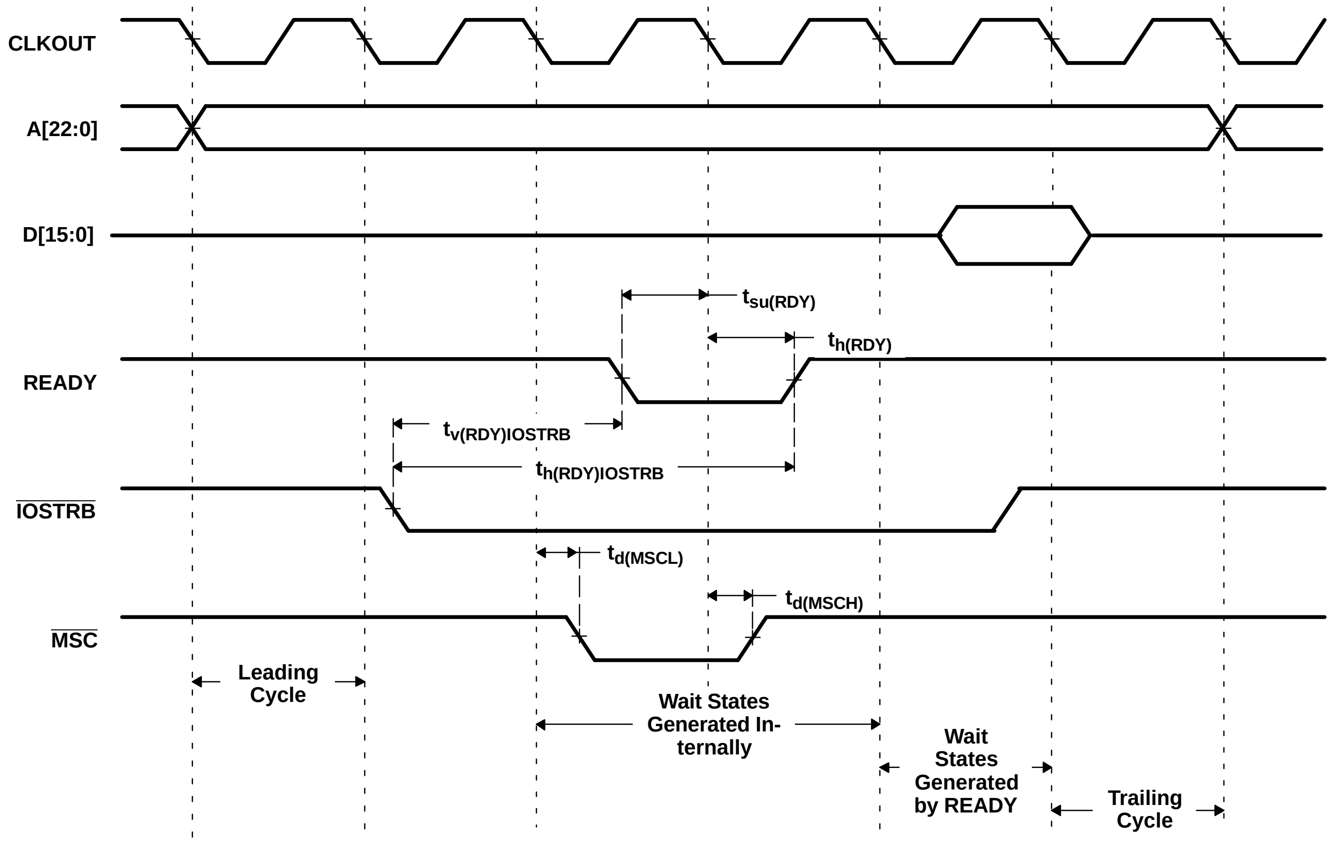



**Figure 5-12. I/O Read With Externally Generated Wait States**

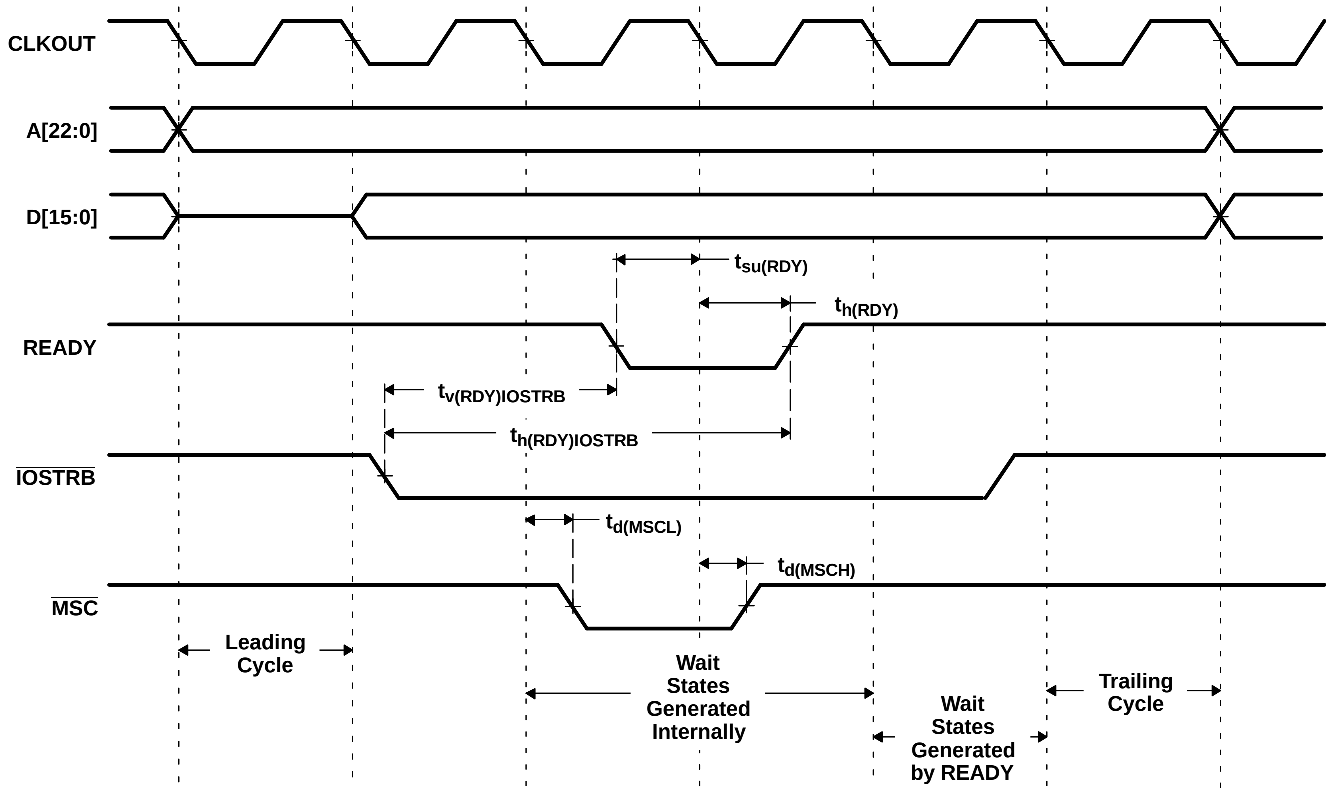



**Figure 5-13. I/O Write With Externally Generated Wait States**

### 5.5.6 $\overline{\text{HOLD}}$ and $\overline{\text{HOLDA}}$ Timings

Table 5-15 and Table 5-16 assume testing over recommended operating conditions and H = $0.5t_{c(CO)}$ (see Figure 5-14).

**Table 5-15. $\overline{\text{HOLD}}$ and $\overline{\text{HOLDA}}$ Timing Requirements**

| | | 5416-120<br>5416-160 | | UNIT |
|---|---|---|---|---|
| | | MIN | MAX | |
| $t_{w(HOLD)}$ | Pulse duration, $\overline{\text{HOLD}}$ low duration | 4H+8 | | ns |
| $t_{su(HOLD)}$ | Setup time, $\overline{\text{HOLD}}$ before CLKOUT low(1) | 7 | | ns |

(1) This input can be driven from an asynchronous source, therefore, there are no specific timing requirments with respect to CLKOUT. However, if this timing is met, the input will be recognized on the CLKOUT edge referenced.

**Table 5-16. $\overline{\text{HOLD}}$ and $\overline{\text{HOLDA}}$ Switching Characteristics**

| PARAMETER | | 5416-120<br>5416-160 | | UNIT |
|---|---|---|---|---|
| | | MIN | MAX | |
| $t_{dis(CLKL-A)}$ | Disable time, Address, $\overline{\text{PS}}$, $\overline{\text{DS}}$, $\overline{\text{IS}}$ high impedance from CLKOUT low | | 3 | ns |
| $t_{dis(CLKL-RW)}$ | Disable time, R/$\overline{\text{W}}$ high impedance from CLKOUT low | | 3 | ns |
| $t_{dis(CLKL-S)}$ | Disable time, $\overline{\text{MSTRB}}$, $\overline{\text{IOSTRB}}$ high impedance from CLKOUT low | | 3 | ns |
| $t_{en(CLKL-A)}$ | Enable time, Address, $\overline{\text{PS}}$, $\overline{\text{DS}}$, $\overline{\text{IS}}$ valid from CLKOUT low | | 2H+3 | ns |
| $t_{en(CLKL-RW)}$ | Enable time, R/$\overline{\text{W}}$ enabled from CLKOUT low | | 2H+3 | ns |
| $t_{en(CLKL-S)}$ | Enable time, $\overline{\text{MSTRB}}$, $\overline{\text{IOSTRB}}$ enabled from CLKOUT low | 2 | 2H+3 | ns |
| $t_{v(HOLDA)}$ | Valid time, $\overline{\text{HOLDA}}$ low after CLKOUT low | – 1 | 4 | ns |
| | Valid time, $\overline{\text{HOLDA}}$ high after CLKOUT low | – 1 | 4 | ns |
| $t_{w(HOLDA)}$ | Pulse duration, $\overline{\text{HOLDA}}$ low duration | 2H–3 | | ns |

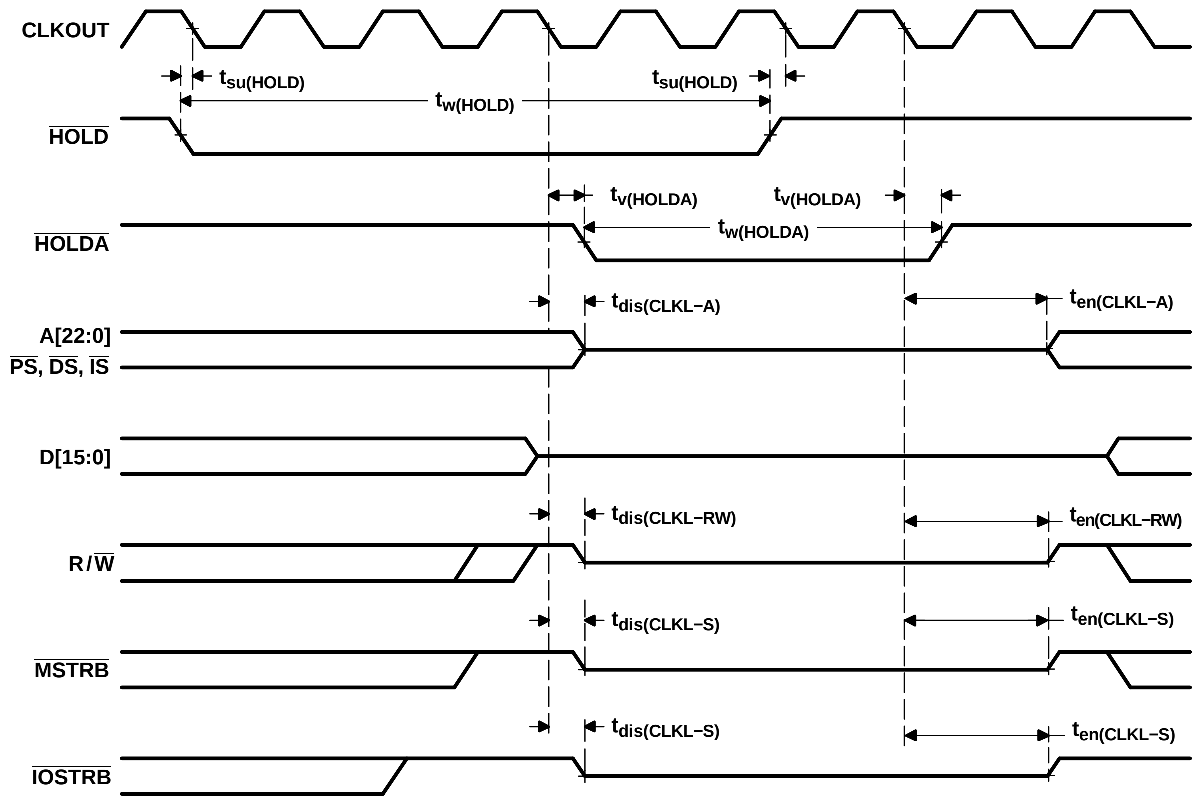



**Figure 5-14. HOLD and HOLDA Timings (HM = 1)**

### 5.5.7 Reset, $\overline{BIO}$, Interrupt, and MP/$\overline{MC}$ Timings

Table 5-17 assumes testing over recommended operating conditions and H = $0.5t_{c(CO)}$ (see Figure 5-15, Figure 5-16, and Figure 5-17).

**Table 5-17. Reset, $\overline{BIO}$, Interrupt, and MP/$\overline{MC}$ Timing Requirements**

| | | 5416-120 5416-160 | | UNIT |
|---|---|---|---|---|
| | | MIN | MAX | |
| $t_{h(RS)}$ | Hold time, $\overline{RS}$ after CLKOUT low(1) | 2 | | ns |
| $t_{h(BIO)}$ | Hold time, $\overline{BIO}$ after CLKOUT low(1) | 4 | | ns |
| $t_{h(INT)}$ | Hold time, $\overline{INTn}$, $\overline{NMI}$, after CLKOUT low(1)(2) | 0 | | ns |
| $t_{h(MPMC)}$ | Hold time, MP/$\overline{MC}$ after CLKOUT low(1) | 4 | | ns |
| $t_{w(RSL)}$ | Pulse duration, $\overline{RS}$ low(3)(4) | 4H + 3 | | ns |
| $t_{w(BIO)S}$ | Pulse duration, $\overline{BIO}$ low, synchronous | 2H + 3 | | ns |
| $t_{w(BIO)A}$ | Pulse duration, $\overline{BIO}$ low, asynchronous | 4H | | ns |
| $t_{w(INTH)S}$ | Pulse duration, $\overline{INTn}$, $\overline{NMI}$ high (synchronous) | 2H + 2 | | ns |
| $t_{w(INTH)A}$ | Pulse duration, $\overline{INTn}$, $\overline{NMI}$ high (asynchronous) | 4H | | ns |
| $t_{w(INTL)S}$ | Pulse duration, $\overline{INTn}$, $\overline{NMI}$ low (synchronous) | 2H + 2 | | ns |
| $t_{w(INTL)A}$ | Pulse duration, $\overline{INTn}$, $\overline{NMI}$ low (asynchronous) | 4H | | ns |
| $t_{w(INTL)WKP}$ | Pulse duration, $\overline{INTn}$, $\overline{NMI}$ low for IDLE2/IDLE3 wakeup | 7 | | ns |
| $t_{su(RS)}$ | Setup time, $\overline{RS}$ before X2/CLKIN low(2)(1) | 3 | | ns |
| $t_{su(BIO)}$ | Setup time, $\overline{BIO}$ before CLKOUT low(1) | 7 | | ns |
| $t_{su(INT)}$ | Setup time, $\overline{INTn}$, $\overline{NMI}$, $\overline{RS}$ before CLKOUT low(1) | 7 | | ns |
| $t_{su(MPMC)}$ | Setup time, MP/$\overline{MC}$ before CLKOUT low(1) | 5 | | ns |

(1) These inputs can be driven from an asynchronous source, therefore, there are no specific timing requirements with respect to CLKOUT, however, if setup and hod timings are met, the input will be recognized on the CLKOUT edge referenced.
(2) The external interrupts ($\overline{INT0}$-$\overline{INT3}$, $\overline{NMI}$) are synchronized to the core CPU by way of a two-flip-flop synchronizer that samples thse inputs with consecutive falling edges of CLKOUT. The input to the interrupt pins is required to represent a1-0-0 sequence at the timing that is corresponding to three CLKOUTs sampling sequence.
(3) If the PLL mode is selected, then at power-on sequence, or at wakeup from IDLE3, $\overline{RS}$ must be held low for at least 50 µs to ensure synchronization and lock-in of the PLL.
(4) Note that $\overline{RS}$ may cause a change in clock frequency, therefore changing the value of H.

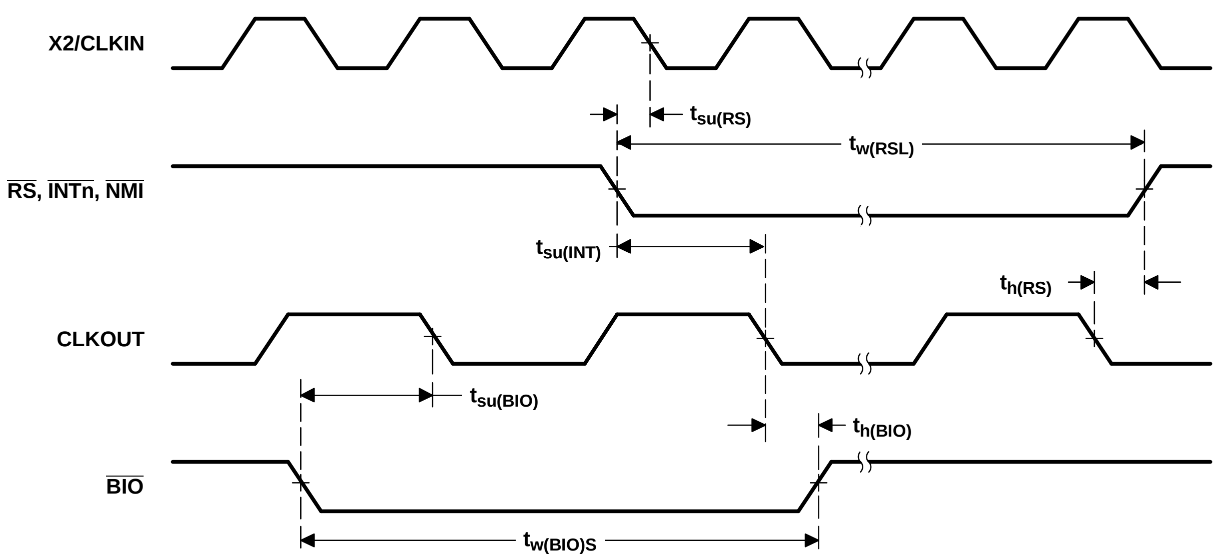



**Figure 5-15. Reset and $\overline{BIO}$ Timings**

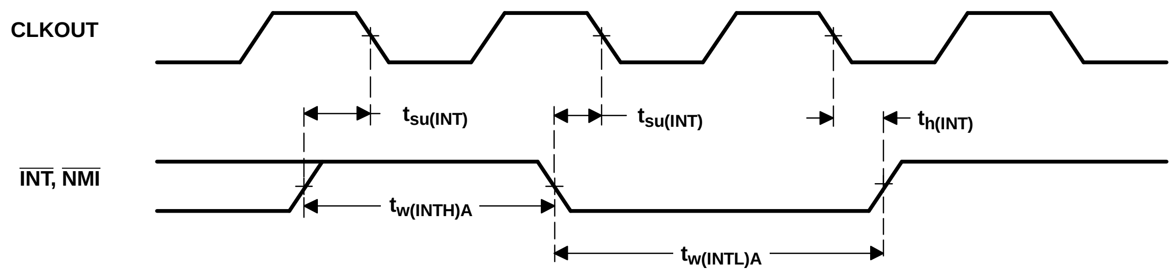



**Figure 5-16. Interrupt Timing**

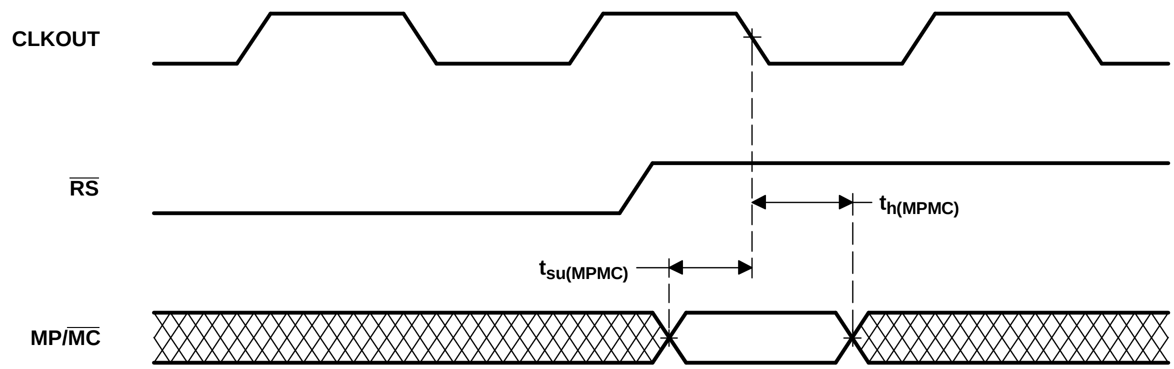



**Figure 5-17. MP/$\overline{MC}$ Timing**

### 5.5.8 Instruction Acquisition (IAQ) and Interrupt Acknowledge (IACK) Timings

Table 5-18 assumes testing over recommended operating conditions and H = $0.5t_{c(CO)}$ (see Figure 5-18).

**Table 5-18. Instruction Acquisition (IAQ) and Interrupt Acknowledge (IACK) Switching Characteristics**

| PARAMETER | | 5416-120 5416-160 MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{d(CLKL-IAQL)}$ | Delay time, CLKOUT low to IAQ low | – 1 | 4 | ns |
| $t_{d(CLKL-IAQH)}$ | Delay time, CLKOUT low to IAQ high | – 1 | 4 | ns |
| $t_{d(CLKL-IACKL)}$ | Delay time, CLKOUT low to IACK low | – 1 | 4 | ns |
| $t_{d(CLKL-IACKH)}$ | Delay time, CLKOUT low to IACK high | – 1 | 4 | ns |
| $t_{d(CLKL-A)}$ | Delay time, CLKOUT low to address valid | – 1 | 4 | ns |
| $t_{w(IAQL)}$ | Pulse duration, IAQ low | 2H – 2 | | ns |
| $t_{w(IACKL)}$ | Pulse duration, IACK low | 2H – 2 | | ns |

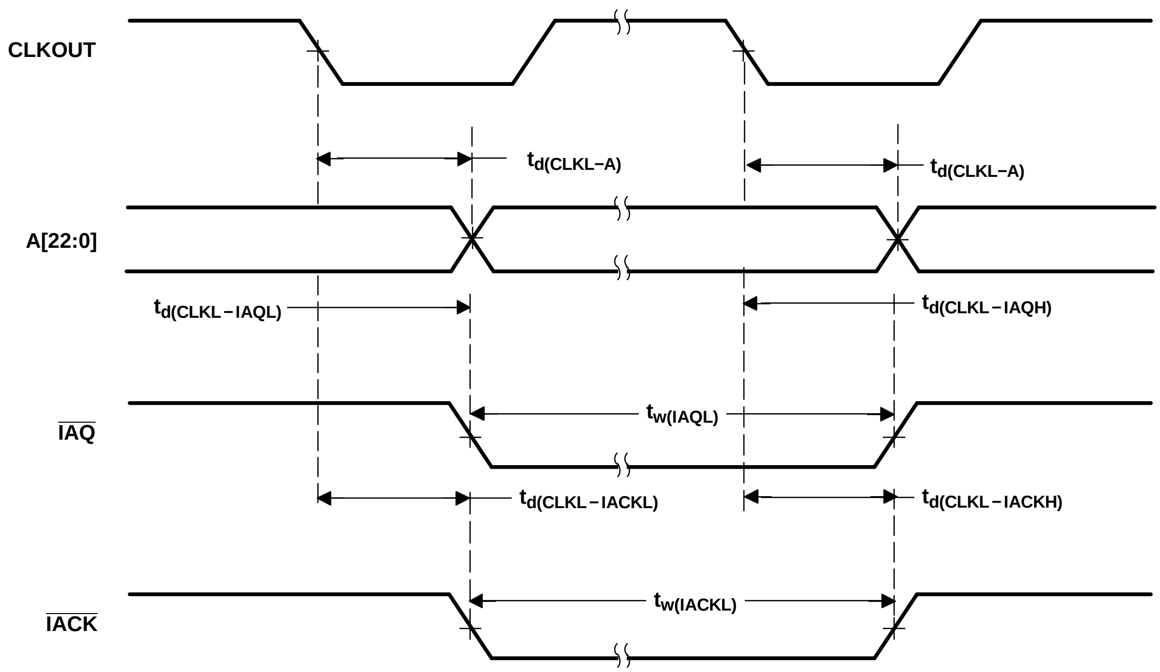



Figure 5-18. Instruction Acquisition (IAQ) and Interrupt Acknowledge (IACK) Timings

### 5.5.9 External Flag (XF) and TOUT Timings

Table 5-19 assumes testing over recommended operating conditions and H = $0.5t_{c(CO)}$ (see Figure 5-19 and Figure 5-20).

**Table 5-19. External Flag (XF) and TOUT Switching Characteristics**

| PARAMETER | | 5416-120<br>5416-160 MIN | MAX | UNIT |
|---|---|---|---|---|
| $t_{d(XF)}$ | Delay time, CLKOUT low to XF high | – 1 | 4 | ns |
| | Delay time, CLKOUT low to XF low | – 1 | 4 | |
| $t_{d(TOUTH)}$ | Delay time, CLKOUT low to TOUT high | – 1 | 4 | ns |
| $t_{d(TOUTL)}$ | Delay time, CLKOUT low to TOUT low | – 1 | 4 | ns |
| $t_{w(TOUT)}$ | Pulse duration, TOUT | 2H – 4 | | ns |

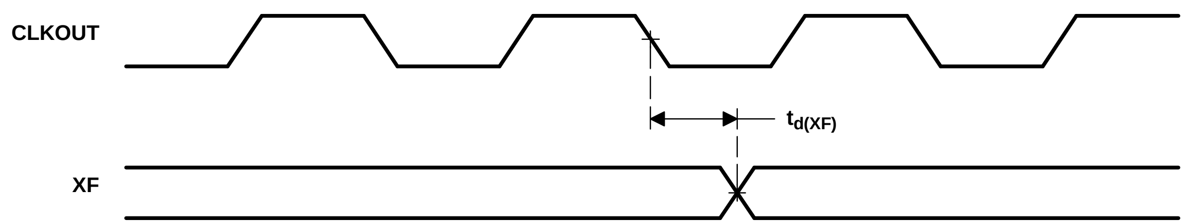



**Figure 5-19. External Flag (XF) Timing**

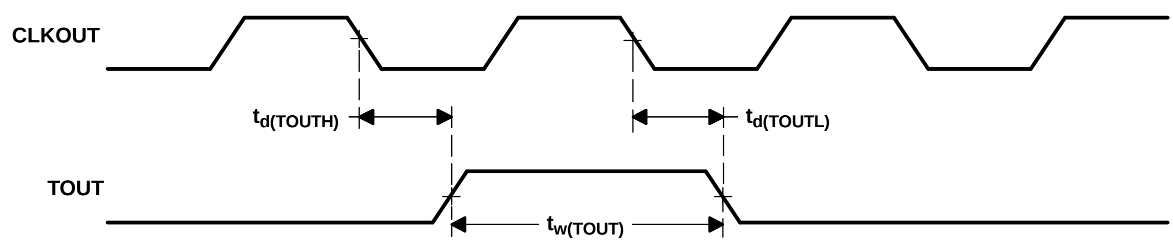



**Figure 5-20. TOUT Timing**

## 5.5.10 Multichannel Buffered Serial Port (McBSP) Timing

### *5.5.10.1 McBSP Transmit and Receive Timings*

Table 5-20 and Table 5-21 assume testing over recommended operating conditions (see Figure 5-21 and Figure 5-22).

**Table 5-20. McBSP Transmit and Receive Timing Requirements**[1]

| | | | 5416-120 5416-160 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| $t_{c(BCKRX)}$ | Cycle time, BCLKR/X[2] | BCLKR/X ext | 4P[3] | | ns |
| $t_{w(BCKRX)}$ | Pulse duration, BCLKR/X high or BCLKR/X low[2] | BCLKR/X ext | 2P–1[3] | | ns |
| $t_{su(BFRH-BCKRL)}$ | Setup time, external BFSR high before BCLKR low | BCLKR int | 8 | | ns |
| | | BCLKR ext | 1 | | |
| $t_{h(BCKRL-BFRH)}$ | Hold time, external BFSR high after BCLKR low | BCLKR int | 1 | | ns |
| | | BCLKR ext | 2 | | |
| $t_{su(BDRV-BCKRL)}$ | Setup time, BDR valid before BCLKR low | BCLKR int | 7 | | ns |
| | | BCLKR ext | 1 | | |
| $t_{h(BCKRL-BDRV)}$ | Hold time, BDR valid after BCLKR low | BCLKR int | 2 | | ns |
| | | BCLKR ext | 3 | | |
| $t_{su(BFXH-BCKXL)}$ | Setup time, external BFSX high before BCLKX low | BCLKX int | 8 | | ns |
| | | BCLKX ext | 1 | | |
| $t_{h(BCKXL-BFXH)}$ | Hold time, external BFSX high after BCLKX low | BCLKX int | 0 | | ns |
| | | BCLKX ext | 2 | | |
| $t_{r(BCKRX)}$ | Rise time, BCKR/X | BCLKR/X ext | | 6 | ns |
| $t_{f(BCKRX)}$ | Fall time, BCKR/X | BCLKR/X ext | | 6 | ns |

(1) CLKRP = CLKXP = FSRP = FSXP = 0. If the polaritiy of any of the signals is inverted, then the timing references of that signal are also inverted.
(2) Note that in some cases, for example when driving another 54x device McBSP, maximum serial port clocking rates may not be achievable at maximum CPU clock frequency due to transmitted data timings and corresponding receive timing requirements. A separate detailed timing analysis should be performend for each specific McBSP interface.
(3) P = 0.5 * processor clock.

### Table 5-21. McBSP Transmit and Receive Switching Characteristics(1)

| PARAMETER | | | | 5416-120 5416-160 MIN | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{c(BCKRX)}$ | Cycle time, BCLKR/X(2) | | BCLKR/X int | 4P(3) | | ns |
| $t_{w(BCKRXH)}$ | Pulse duration, BCLKR/X high(2) | | BCLKR/X int | D – 1(4) | D + 1(4) | ns |
| $t_{w(BCKRXL)}$ | Pulse duration, BCLKR/X low(2) | | BCLKR/X int | C – 1(4) | C + 1(4) | ns |
| $t_{d(BCKRH-BFRV)}$ | Delay time, BCLKR high to internal BFSR valid | | BCLKR int | – 3 | 3 | ns |
| | | | BCLKR ext | 0 | 6 | ns |
| $t_{d(BCKXH-BFXV)}$ | Delay time, BCLKX high to internal BFSX valid | | BCLKX int | – 1 | 5 | ns |
| | | | BCLKX ext | 3 | 11 | |
| $t_{dis(BCKXH-BDXHZ)}$ | Disable time, BCLKX high to BDX high impedance following last data bit of transfer | | BCLKX int | | 6 | ns |
| | | | BCLKX ext | | 10 | |
| $t_{d(BCKXH-BDXV)}$ | Delay time, BCLKX high to BDX valid | DXENA = 0 | BCLKX int | – 1(5) | 10 | ns |
| | | | BCLKX ext | 3 | 20 | |
| | | DXENA = 1 | BCLKX int | – 1(5) | 20 | |
| | | | BCLKX ext | 3 | 30 | |
| $t_{d(BFXH-BDXV)}$ | Delay time, BFSX high to BDX valid ONLY applies when in data delay 0 (XDATDLY = 00b) mode | | BFSX int | –1(5) | 7 | ns |
| | | | BFSX ext | 3 | 11 | |

(1) CLKRP = CLKXP = FSRP = FSXP = 0. If the polaritiy of any of the signals is inverted, then the timing references of that signal are also inverted.
(2) Note that in some cases, for example when driving another 54x device McBSP, maximum serial port clocking rates may not be achievable at maximum CPU clock frequency due to transmitted data timings and corresponding receive timing requirements. A separate detailed timing analysis should be performend for each specific McBSP interface.
(3) P = 0.5 * processor clock.
(4) T = BCLKRX period = (1 + CLKGDV) * 2P
C = BCLKRX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) * 2P when CLKGDV is even
D = BCLKRX high pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2 +1) * 2P when CLKGDV is even
(5) Minimum delay times also represent minimum output hold times.

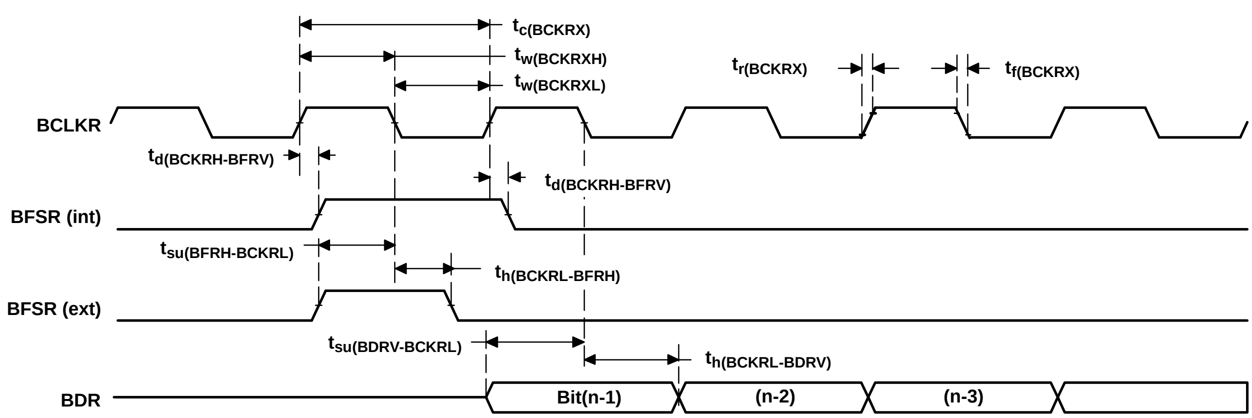



Figure 5-21. McBSP Receive Timings

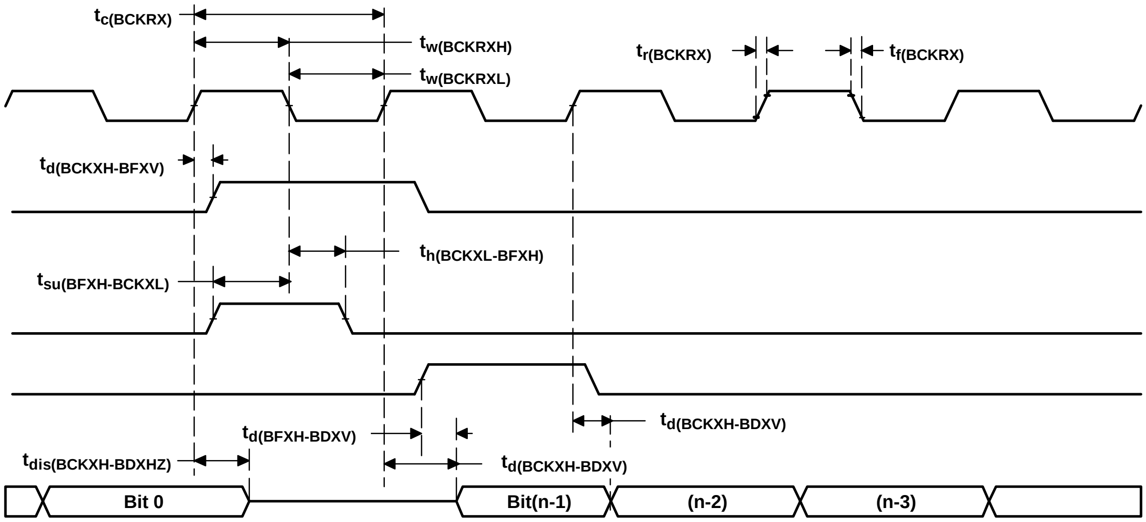



**Figure 5-22. McBSP Transmit Timings**

### 5.5.10.2 McBSP General-Purpose I/O Timing

Table 5-22 and Table 5-23 assume testing over recommended operating conditions (see Figure 5-23).

**Table 5-22. McBSP General-Purpose I/O Timing Requirements**

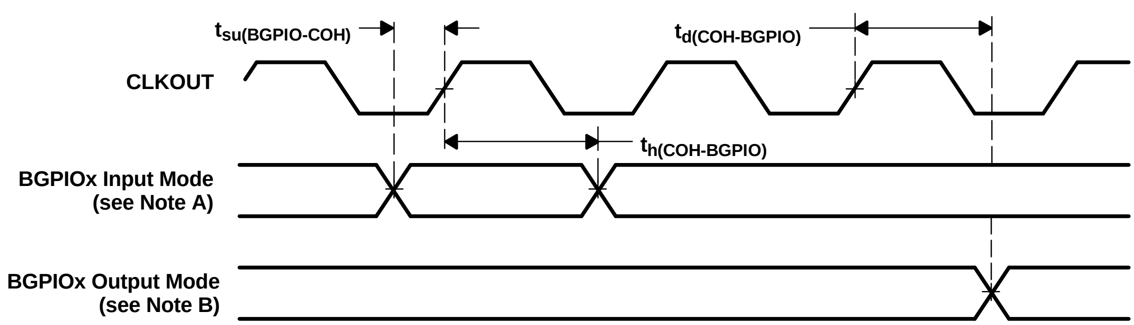
| | | 5416-120<br>5416-160 MIN | 5416-120<br>5416-160 MAX | UNIT |
|---|---|---|---|---|
| $t_{su(BGPIO-COH)}$ | Setup time, BGPIOx input mode before CLKOUT high(1) | 7 | | ns |
| $t_{h(COH-BGPIO)}$ | Hold time, BGPIOx input mode after CLKOUT high(1) | 0 | | ns |

(1) BGPIOx refers to BCLKRx, BFSRx, BDRx, BCLKXx, or BFSXx when configured as a general-purpose input.

**Table 5-23. McBSP General-Purpose I/O Switching Characteristics**

| PARAMETER | | 5416-120<br>5416-160 MIN | 5416-120<br>5416-160 MAX | UNIT |
|---|---|---|---|---|
| $t_{d(COH-BGPIO)}$ | Delay time, CLKOUT high to BGPIOx output mode(1) | – 2 | 4 | ns |

(1) BGPIOx refers to BCLKRx, BFSRx, BDRx, BCLKXx, or BFSXx when configured as a general-purpose output.




A. BGPIOx refers to BCLKRx, BFSRx, BDRx, BCLKXx, or BFSXx when configured as a general-purpose input.
B. BGPIOx refers to BCLKRx, BFSRx, BCLKXx, BFSXx, or BDXx when configured as a general-purpose output.

**Figure 5-23. McBSP General-Purpose I/O Timings**

### 5.5.10.3 McBSP as SPI Master or Slave Timing

Table 5-24 to Table 5-31 assume testing over recommended operating conditions (see Figure 5-24, Figure 5-25, Figure 5-26, and Figure 5-27).

**Table 5-24. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 10b, CLKXP = 0)(1)**

| | | 5416-120<br>5416-160 | | | | UNIT |
|---|---|---|---|---|---|---|
| | | MASTER | | SLAVE | | |
| | | MIN | MAX | MIN | MAX | |
| $t_{su(BDRV-BCKXL)}$ | Setup time, BDR valid before BCLKX low | 12 | | 2 – 6P(2) | | ns |
| $t_{h(BCKXL-BDRV)}$ | Hold time, BDR valid after BCLKX low | 4 | | 5 + 12P(2) | | ns |

(1) For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
(2) P = 0.5 * processor clock.

**Table 5-25. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 10b, CLKXP = 0)(1)**

| PARAMETER | | 5416-120<br>5416-160 | | | | UNIT |
|---|---|---|---|---|---|---|
| | | MASTER(2) | | SLAVE | | |
| | | MIN | MAX | MIN | MAX | |
| $t_{h(BCKXL-BFXL)}$ | Hold time, BFSX low after BCLKX low(3) | T – 3 | T + 4 | | | ns |
| $t_{d(BFXL-BCKXH)}$ | Delay time, BFSX low to BCLKX high(4) | C – 4 | C + 3 | | | ns |
| $t_{d(BCKXH-BDXV)}$ | Delay time, BCLKX high to BDX valid | – 4 | 5 | 6P + 2(5) | 10P + 17(5) | ns |
| $t_{dis(BCKXL-BDXHZ)}$ | Disable time, BDX high impedance following last data bit from BCLKX low | C – 2 | C + 3 | | | ns |
| $t_{dis(BFXH-BDXHZ)}$ | Disable time, BDX high impedance following last data bit from BFSX high | | | 2P– 4(5) | 6P + 17(5) | ns |
| $t_{d(BFXL-BDXV)}$ | Delay time, BFSX low to BDX valid | | | 4P+ 2(5) | 8P + 17(5) | ns |

(1) For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
(2) T = BCLKX period = (1 + CLKGDV) * 2P
C = BCLKX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) *2P when CLKGDV is even
(3) FSRP = FSXP = 1. As a SPI master, BFSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on BFSX and BFSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FRSRM = 2 for master McBSP
CLKXM = CLKRm = FSXM = FSRM = 0 for slave McBSP
(4) BFSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (BCLKX).
(5) P = 0.5 * processor clock.

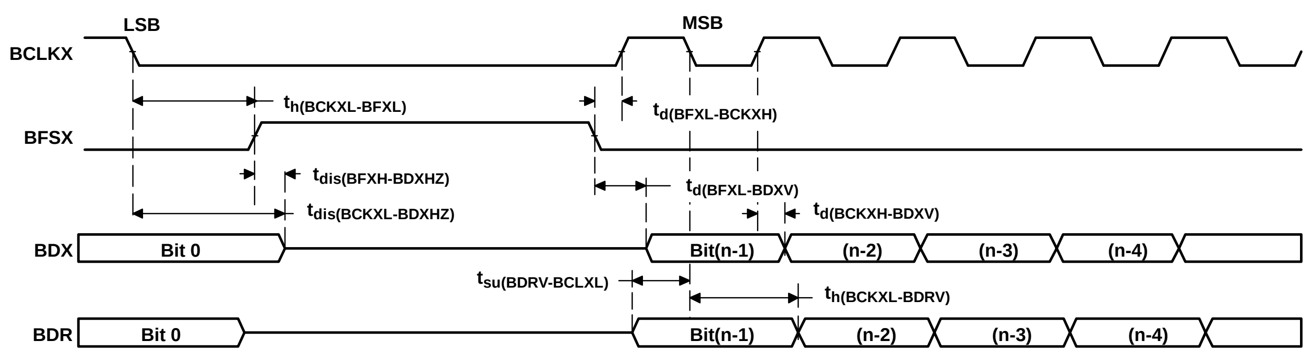



**Figure 5-24. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0**

### Table 5-26. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 11b, CLKXP = 0)[1]

| | | 5416-120<br>5416-160 | | | | UNIT |
|---|---|---|---|---|---|---|
| | | MASTER | | SLAVE | | |
| | | MIN | MAX | MIN | MAX | |
| $t_{su(BDRV-BCKXL)}$ | Setup time, BDR valid before BCLKX low | 12 | | 2 – 6P[2] | | ns |
| $t_{h(BCKXH-BDRV)}$ | Hold time, BDR valid after BCLKX high | 4 | | 5 + 12P[2] | | ns |

(1) For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
(2) P = 0.5 * processor clock.

### Table 5-27. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 11b, CLKXP = 0)[1]

| PARAMETER | | 5416-120<br>5416-160 | | | | UNIT |
|---|---|---|---|---|---|---|
| | | MASTER[2] | | SLAVE | | |
| | | MIN | MAX | MIN | MAX | |
| $t_{h(BCKXL-BFXL)}$ | Hold time, BFSX low after BCLKX low[3] | C –3 | C + 4 | | | ns |
| $t_{d(BFXL-BCKXH)}$ | Delay time, BFSX low to BCLKX high[4] | T – 4 | T + 3 | | | ns |
| $t_{d(BCKXL-BDXV)}$ | Delay time, BCLKX low to BDX valid | – 4 | 5 | 6P + 2[5] | 10P + 17[5] | ns |
| $t_{dis(BCKXL-BDXHZ)}$ | Disable time, BDX high impedance following last data bit from BCLKX low | – 2 | 4 | 6P – 4[5] | 10P + 17[5] | ns |
| $t_{d(BFXL-BDXV)}$ | Delay time, BFSX low to BDX valid | D – 2 | D + 4 | 4P + 2[5] | 8P + 17[5] | ns |

(1) For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
(2) T = BCLKX period = (1 + CLKGDV) * 2P
C = BCLKX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) *2P when CLKGDV is even
D = BCLKX high pulse width = T/2 when CLKGDV is odd or zero and (CLKGDV/2 + 1) * 2P when CLKGDV is even
(3) FSRP = FSXP = 1. As a SPI master, BFSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on BFSX and BFSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FRSRM = 2 for master McBSP
CLKXM = CLKRm = FSXM = FSRM = 0 for slave McBSP
(4) BFSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (BCLKX).
(5) P = 0.5 * processor clock.

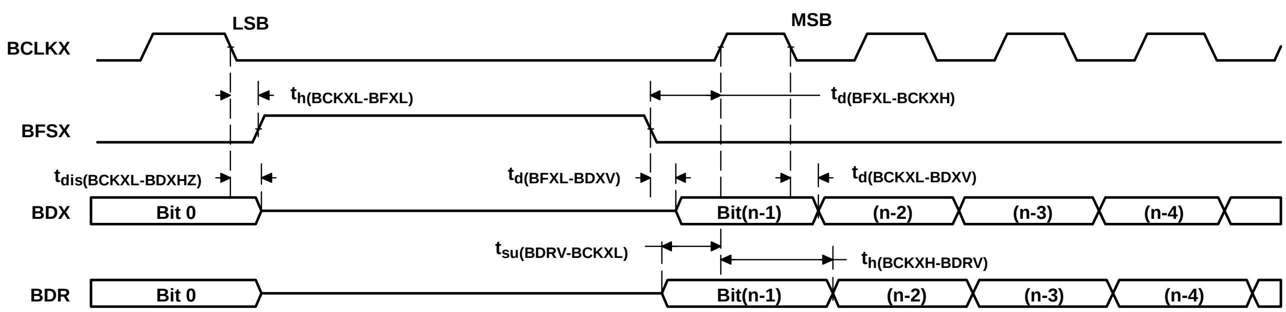



Figure 5-25. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

### Table 5-28. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 10b, CLKXP = 1)(1)

| | | 5416-120 5416-160 | | | | UNIT |
|---|---|---|---|---|---|---|
| | | MASTER | | SLAVE | | |
| | | MIN | MAX | MIN | MAX | |
| $t_{su(BDRV-BCKXH)}$ | Setup time, BDR valid before BCLKX high | 12 | | 2 – 6P(2) | | ns |
| $t_{h(BCKXH-BDRV)}$ | Hold time, BDR valid after BCLKX high | 4 | | 5 + 12P(2) | | ns |

(1) For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
(2) P = 0.5 * processor clock.

### Table 5-29. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 10b, CLKXP = 1)(1)

| PARAMETER | | 5416-120 5416-160 | | | | UNIT |
|---|---|---|---|---|---|---|
| | | MASTER(2) | | SLAVE | | |
| | | MIN | MAX | MIN | MAX | |
| $t_{h(BCKXH-BFXL)}$ | Hold time, BFSX low after BCLKX high(3) | T – 3 | T + 4 | | | ns |
| $t_{d(BFXL-BCKXL)}$ | Delay time, BFSX low to BCLKX lowTNote9543(4) | D – 4 | D + 3 | | | ns |
| $t_{d(BCKXL-BDXV)}$ | Delay time, BCLKX low to BDX valid | – 4 | 5 | 6P + 2(5) | 10P + 17(5) | ns |
| $t_{dis(BCKXH-BDXHZ)}$ | Disable time, BDX high impedance following last data bit from BCLKX high | D – 2 | D + 3 | | | ns |
| $t_{dis(BFXH-BDXHZ)}$ | Disable time, BDX high impedance following last data bit from BFSX high | | | 2P – 4(5) | 6P + 17(5) | ns |
| $t_{d(BFXL-BDXV)}$ | Delay time, BFSX low to BDX valid | | | 4P + 2(5) | 8P + 17(5) | ns |

(1) For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
(2) T = BCLKX period = (1 + CLKGDV) * 2P
D = BCLKX high pulse width = T/2 when CLKGDV is odd or zero and (CLKGDV/2 + 1) * 2P when CLKGDV is even
(3) FSRP = FSXP = 1. As a SPI master, BFSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on BFSX and BFSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FRSRM = 2 for master McBSP
CLKXM = CLKRm = FSXM = FSRM = 0 for slave McBSP
(4) BFSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (BCLKX).
(5) P = 0.5 * processor clock.

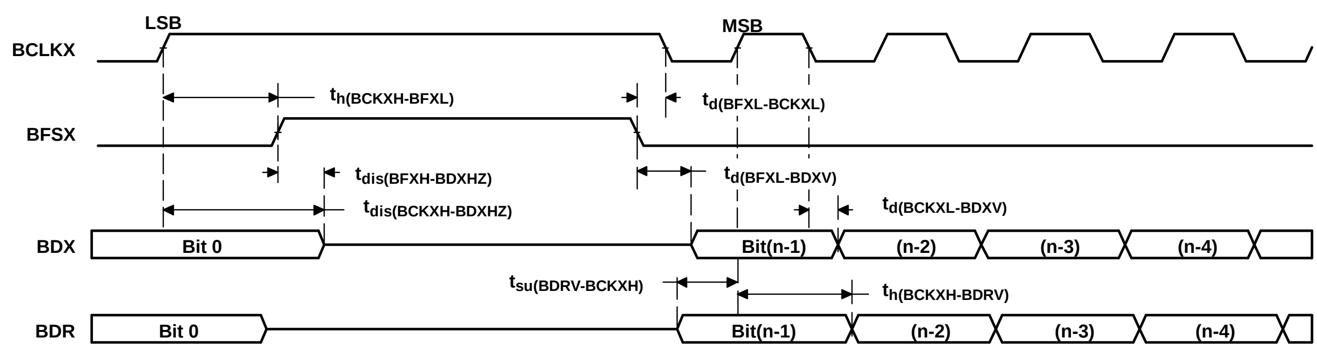



Figure 5-26. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

### Table 5-30. McBSP as SPI Master or Slave Timing Requirements (CLKSTP = 11b, CLKXP = 1)(1)

| | | 5416-120<br>5416-160 | | | | UNIT |
|---|---|---|---|---|---|---|
| | | MASTER | | SLAVE | | |
| | | MIN | MAX | MIN | MAX | |
| $t_{su(BDRV-BCKXL)}$ | Setup time, BDR valid before BCLKX low | 12 | | 2 – 6P(2) | | ns |
| $t_{h(BCKXL–BDRV)}$ | Hold time, BDR valid after BCLKX low | 4 | | 5 + 12P(2) | | ns |

(1) For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
(2) P = 0.5 * processor clock.

### Table 5-31. McBSP as SPI Master or Slave Switching Characteristics (CLKSTP = 11b, CLKXP = 1)(1)

| PARAMETER | | 5416-120<br>5416-160 | | | | UNIT |
|---|---|---|---|---|---|---|
| | | MASTER(2) | | SLAVE | | |
| | | MIN | MAX | MIN | MAX | |
| $t_{h(BCKXH-BFXL)}$ | Hold time, BFSX low after BCLKX high(3) | D – 3 | D + 4 | | | ns |
| $t_{d(BFXL-BCKXL)}$ | Delay time, BFSX low to BCLKX low(4) | T – 4 | T + 3 | | | ns |
| $t_{d(BCKXH-BDXV)}$ | Delay time, BCLKX high to BDX valid | – 4 | 5 | 6P + 2(5) | 10P + 17(5) | ns |
| $t_{dis(BCKXH-BDXHZ)}$ | Disable time, BDX high impedance following last data bit from BCLKX high | – 2 | 4 | 6P – 4(5) | 10P + 17(5) | ns |
| $t_{d(BFXL-BDXV)}$ | Delay time, BFSX low to BDX valid | C – 2 | C + 4 | 4P + 2(5) | 8P + 17(5) | ns |

(1) For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
(2) T = BCLKX period = (1 + CLKGDV) * 2P
C = BCLKX low pulse width = T/2 when CLKGDV is odd or zero and = (CLKGDV/2) * 2P when CLKGDV is even
D = BCLKX high pulse width = T/2 when CLKGDV is odd or zero and (CLKGDV/2 + 1) * 2P when CLKGDV is even
(3) FSRP = FSXP = 1. As a SPI master, BFSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on BFSX and BFSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FRSRM = 2 for master McBSP
CLKXM = CLKRm = FSXM = FSRM = 0 for slave McBSP
(4) BFSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (BCLKX).
(5) P = 0.5 * processor clock.

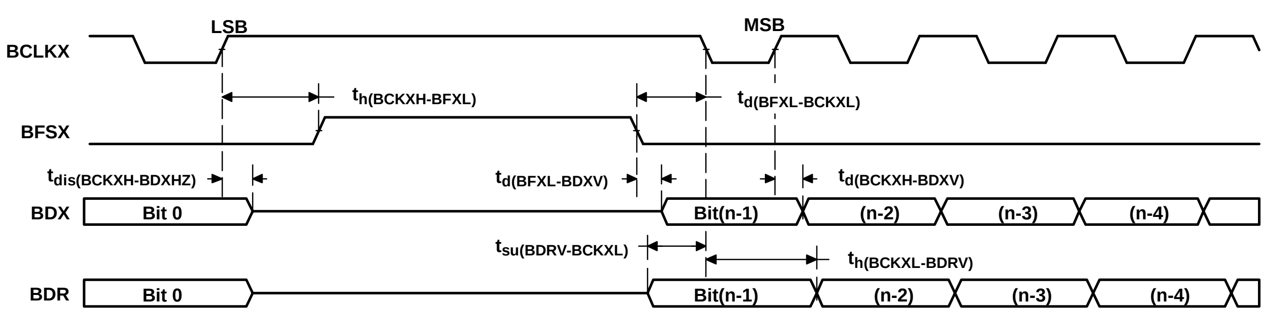



**Figure 5-27. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1**

### 5.5.11 Host-Port Interface Timing

#### *5.5.11.1 HPI8 Mode*

Table 5-32 and Table 5-33 assume testing over recommended operating conditions and P = 0.5 * processor clock (see Figure 5-28 through Figure 5-31). In the following tables, DS refers to the logical OR of $\overline{HCS}$, $\overline{HDS1}$, and $\overline{HDS2}$. HD refers to any of the HPI data bus pins (HD0, HD1, HD2, etc.). HAD stands for HCNTL0, HCNTL1, and HR/$\overline{W}$.

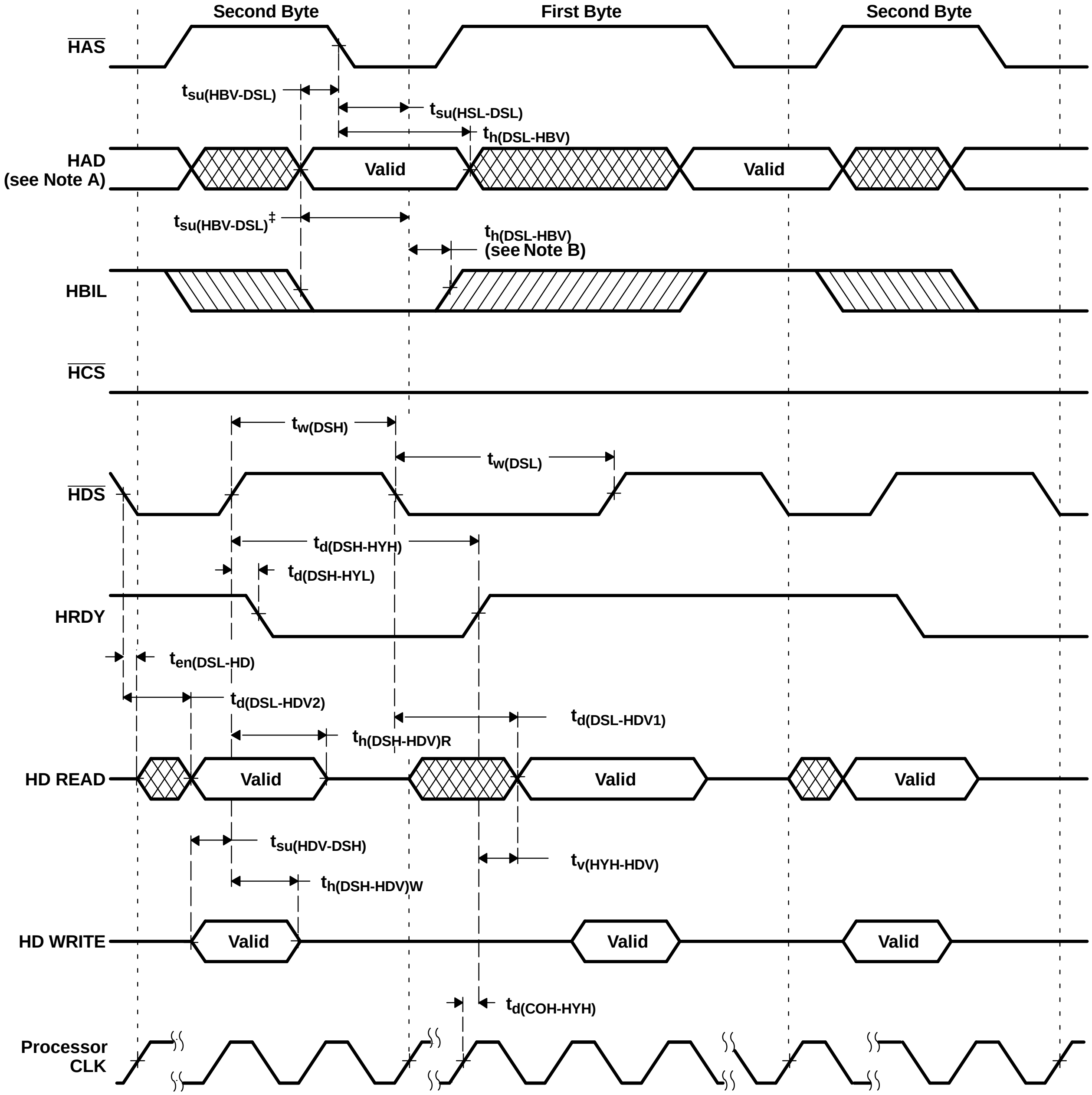
**Table 5-32. HPI8 Mode Timing Requirements**

| | | 5416-120<br>5416-160 | | UNIT |
|---|---|---|---|---|
| | | MIN | MAX | |
| $t_{su(DSL-HBV)}$ | Setup time, HBIL and HAD valid before DS low (when $\overline{HAS}$ is not used), or HBIL and HAD valid before $\overline{HAS}$ low | 6 | | ns |
| $t_{h(DSL-HBV)}$ | Hold time, HBIL and HAD valid after DS low (when $\overline{HAS}$ is not used), or HBIL and HAD valid after $\overline{HAS}$ low | 3 | | ns |
| $t_{su(HSL-DSL)}$ | Setup time, $\overline{HAS}$ low before DS low | 8 | | ns |
| $t_{w(DSL)}$ | Pulse duration, DS low | 13 | | ns |
| $t_{w(DSH)}$ | Pulse duration, DS high | 7 | | ns |
| $t_{su(HDV-DSH)}$ | Setup time, HD valid before DS high, HPI write | 3 | | ns |
| $t_{h(DSH-HDV)W}$ | Hold time, HD valid after DS high, HPI write | 2 | | ns |
| $t_{su(GPIO-COH)}$ | Setup time, HDx input valid before CLKOUT high, HDx configured as general-purpose input | 3 | | ns |
| $t_{h(GPIO-COH)}$ | Hold time, HDx input valid before CLKOUT high, HDx configured as general-purpose input | 0 | | ns |

## Table 5-33. HPI8 Mode Switching Characteristics

| PARAMETER | | | 5416-120 5416-160 MIN | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{en(DSL-HD)}$ | Enable time, HD driven from DS low | | 0 | 10 | ns |
| $t_{d(DSL-HDV1)}$ | Delay time, DS low to HD valid for first byte of an HPI read | Case 1a: Memory accesses when DMAC is active in 32-bit mode and $t_{w(DSH)} < 36P$ (1) | | $36P + 10 - t_{w(DSH)}$ | ns |
| | | Case 1b: Memory accesses when DMAC is active in 32-bit mode and $t_{w(DSH)} \geq 36P$ (1) | | 10 | |
| | | Case 1c: Memory accesses when DMAC is active in 16-bit mode and $t_{w(DSH)} < 18P$ (1) | | $18P + 10 - t_{w(DSH)}$ | |
| | | Case 1d: Memory accesses when DMAC is active in 16-bit mode and $t_{w(DSH)} \geq 18P$ (1) | | 10 | |
| | | Case 2a: Memory accesses when DMAC is inactive and $t_{w(DSH)} < 10P$ (1) | | $10P + 10 - t_{w(DSH)}$ | |
| | | Case 2b: Memory accesses when DMAC is inactive and $t_{w(DSH)} \geq 10P$ (1) | | 10 | |
| | | Case 3: Register accesses | | 10 | |
| $t_{d(DSL-HDV2)}$ | Delay time, DS low to HD valid for second byte of an HPI read | | | 10 | ns |
| $t_{h(DSH-HDV)R}$ | Hold time, HD valid after DS high, for a HPI read | | 0 | | ns |
| $t_{v(HYH-HDV)}$ | Valid time, HD valid after HRDY high | | | 2 | ns |
| $t_{d(DSH-HYL)}$ | Delay time, DS high to HRDY low (2) | | | 8 | ns |
| $t_{d(DSH-HYH)}$ | Delay time, DS high to HRDY high (2) | Case 1a: Memory accesses when DMAC is active in 16-bit mode (1) | | 18P + 6 | ns |
| | | Case 1b: Memory accesses when DMAC is active in 32-bit mode (1) | | 36P + 6 | |
| | | Case 2: Memory accesses when DMAC is inactive (1) | | 10P + 6 | |
| | | Case 3: Write accesses to HPIC register (3) | | 6P + 6 | |
| $t_{d(HCS-HRDY)}$ | Delay time, $\overline{HCS}$ low/high to HRDY low/high | | | 6 | ns |
| $t_{d(COH-HYH)}$ | Delay time, CLKOUT high to HRDY high | | | 9 | ns |
| $t_{d(COH-HTX)}$ | Delay time, CLKOUT high to $\overline{HINT}$ change | | | 6 | ns |
| $t_{d(COH-GPIO)}$ | Delay time, CLKOUT high to HDx output change. HDx is configured as a general-purpose output | | | 5 | ns |

(1) DMAS stands for direct memory access controller. The HPI8 shares the internal DMA bus with the DMAC, thus HPI8 access times are affected by DMAC activity.
(2) The HRDY output is always high when the $\overline{HCS}$ input is high, regardless of DS timings.
(3) This timing applies when writing a one to the DSPINT bit or $\overline{HINT}$ bit of the HPIC register. All other writes to the HPIC occur asynchronously, and do not cause HRDY to be deasserted.




A. HAD refers to HCNTL0, HCNTL1, and HR/$\overline{\text{W}}$.

B. When $\overline{\text{HAS}}$ is not used ($\overline{\text{HAS}}$ always high)

**Figure 5-28. Using $\overline{\text{HDS}}$ to Control Accesses ($\overline{\text{HCS}}$ Always Low)**

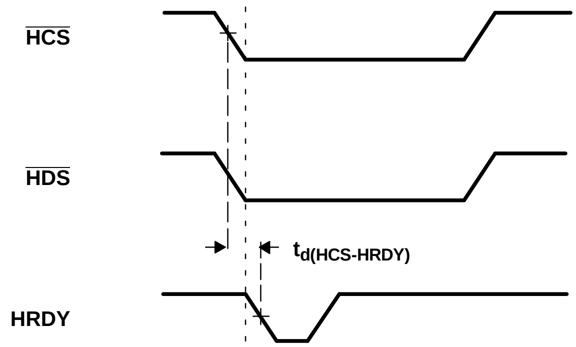



**Figure 5-29. Using HCS to Control Accesses**

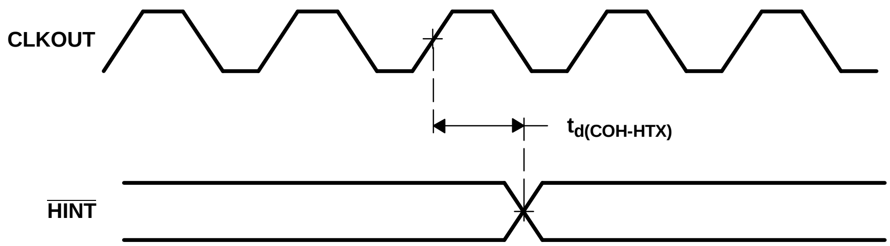



**Figure 5-30. HINT Timing**

In Figure 5-31, GPIOx refers to HD0, HD1, HD2, ...HD7, when the HD bus is configured for general-purpose input/output (I/O).

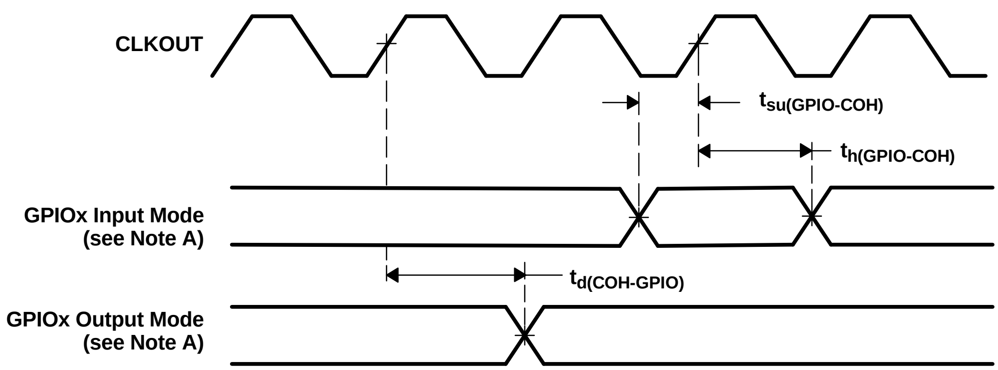



**Figure 5-31. GPIOx Timings**

### 5.5.11.2 HPI16 Mode

Table 5-34 and Table 5-35 assume testing over recommended operating conditions and P = 0.5 * processor clock (see Figure 5-32 through Figure 5-34). In the following tables, DS refers to the logical OR of $\overline{HCS}$, $\overline{HDS1}$, and $\overline{HDS2}$, and HD refers to any of the HPI data bus pins (HD0, HD1, HD2, etc.). These timings are shown assuming that $\overline{HDS}$ is the signal controlling the transfer. See the *TMS320C54x DSP Reference Set,Volume 5: Enhanced Peripherals* (literature number SPRU302) for additional information.

**Table 5-34. HPI16 Mode Timing Requirements**

| | | | | 5416-120<br>5416-160 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| $t_{su(HBV-DSL)}$ | Setup time, HR/$\overline{W}$ valid before DS falling edge | | | 6 | | ns |
| $t_{h(DSL-HBV)}$ | Hold time, HR/$\overline{W}$ valid after DS falling edge | | | 5 | | ns |
| $t_{su(HAV-DSH)}$ | Setup time, address valid before DS rising edge (write) | | | 5 | | ns |
| $t_{su(HAV-DSL)}$ | Setup time, address valid before DS falling edge (read) | | | –(4P– 6)(1) | | ns |
| $t_{h(DSH-HAV)}$ | Hold time, address valid after DS rising edge | | | 1 | | ns |
| $t_{w(DSL)}$ | Pulse duration, DS low | | | 30 | | ns |
| $t_{w(DSH)}$ | Pulse duration, DS high | | | 10 | | ns |
| $t_{c(DSH-DSH)}$ | Cycle time, DS rising edge to next DS rising edge | Memory accesses with no DMA activity. | Reads | 10P + 30(1) | | ns |
| | | | Writes | 10P + 10(1) | | |
| | | Memory accesses with 16-bit DMA activity. | Reads | 16P + 30(1) | | |
| | | | Writes | 16P + 10(1) | | |
| | | Memory accesses with 32-bit DMA activity. | Reads | 24P + 30(1) | | |
| | | | Writes | 24P + 10(1) | | |
| $t_{su(HDV-DSH)W}$ | Setup time, HD valid before DS rising edge | | | 8 | | ns |
| $t_{h(DSH-HDV)W}$ | Hold time, HD valid after DS rising edge, write | | | 2 | | ns |

(1) P = 0.5 * processor clock.

## Table 5-35. HPI16 Mode Switching Characteristics

| PARAMETER | | | 5416-120 5416-160 MIN | 5416-120 5416-160 MAX | UNIT |
|---|---|---|---|---|---|
| $t_{d(DSL-HDD)}$ | Delay time, DS low to HD driven | | 0 | 10 | ns |
| $t_{d(DSL-HDV1)}$ | Delay time, DS low to HD valid for first word of an HPI read | Case 1a: Memory accesses initiated immediately following a write when DMAC is active in 32-bit mode and $t_{w(DSH)}$ was < 26P | | $48P + 20 - t_{w(DSH)}$ | ns |
| | | Case 1b: Memory access not immediately following a write when DMAC is active in 32-bit mode | | 24P + 20 | ns |
| | | Case 1c: Memory accesses initiated immediately following a write when DMAC is active in 16-bit mode and $t_{w(DSH)}$ was < 18P | | $32P + 20 - t_{w(DSH)}$ | ns |
| | | Case 1d: Memory accesses not immediately following a write when DMAC is active in 16-bit mode | | 16P + 20 | ns |
| | | Case 2a: Memory accesses initiated immediately following a write when DMAC is inactive and $t_{w(DSH)}$ was < 10P | | $20P + 20 - t_{w(DSH)}$ | ns |
| | | Case 2b: Memory accesses not immediately following a write when DMAC is inactive | | 10P + 20 | ns |
| $t_{d(DSH-HYH)}$ | Delay time, DS high to HRDY high | Memory writes when no DMA is active | | 10P + 5 | ns |
| | | Memory writes with one or more 16-bit DMA channels active | | 16P + 5 | ns |
| | | Memory writes with one or more 32-bit DMA channels active | | 24P + 5 | ns |
| $t_{v(HYH-HDV)}$ | Valid time, HD valid after HRDY high | | | 7 | ns |
| $t_{h(DSH-HDV)R}$ | Hold time, HD valid after DS rising edge, read | | 1 | 6 | ns |
| $t_{d(COH-HYH)}$ | Delay time, CLKOUT rising edge to HRDY high | | | 5 | ns |
| $t_{d(DSL-HYL)}$ | Delay time, DS low to HRDY low | | | 12 | ns |
| $t_{d(DSH-HYL)}$ | Delay time, DS high to HRDY low | | | 12 | ns |

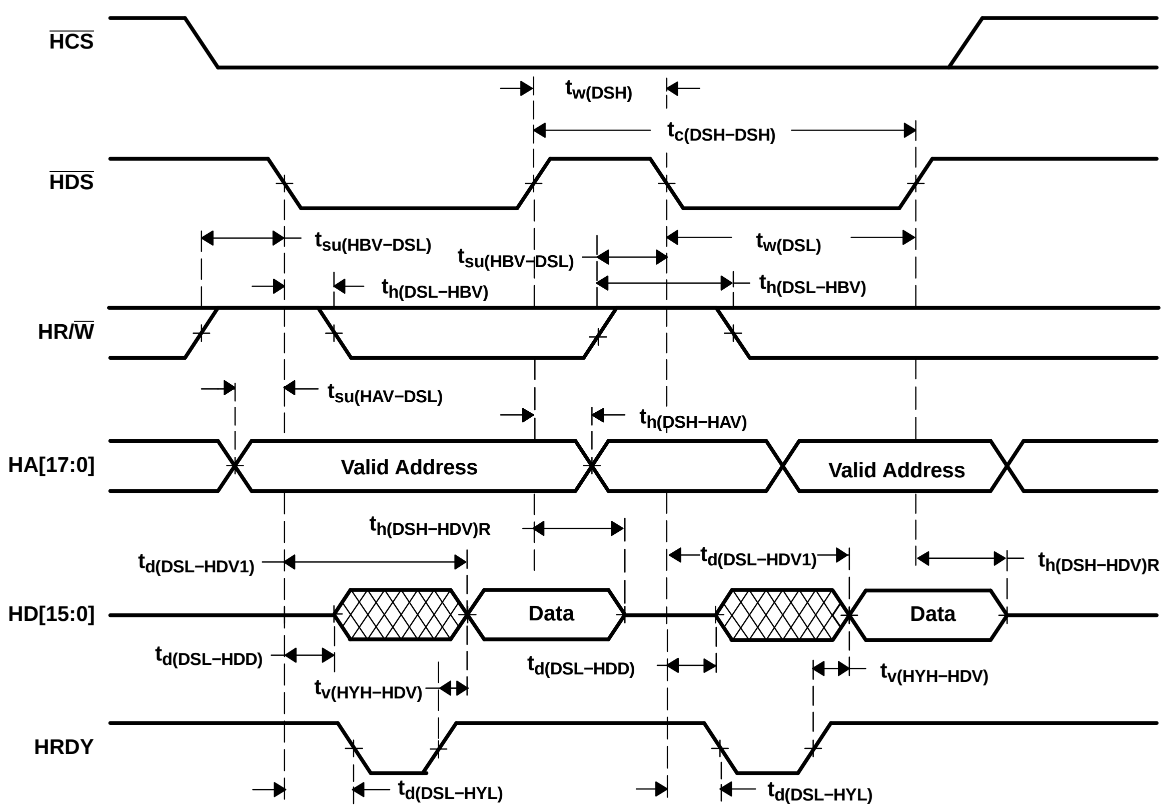



Figure 5-32. Nonmultiplexed Read Timings

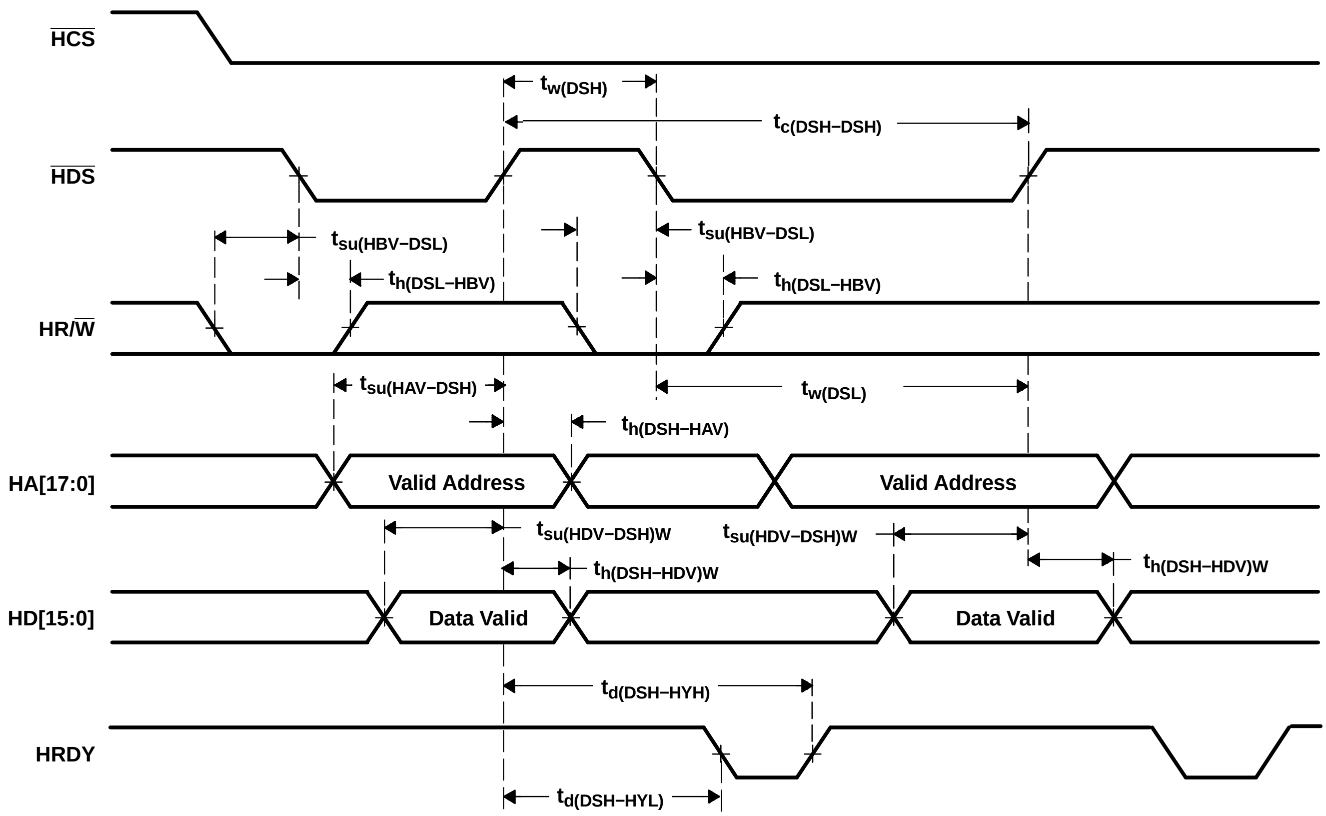



**Figure 5-33. Nonmultiplexed Write Timings**

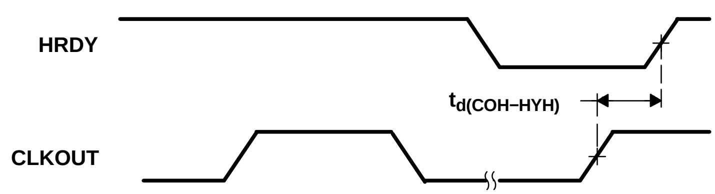



**Figure 5-34. HRDY Relative to CLKOUT**

# 6 Mechanical Data

The following mechanical package diagram(s) reflect the most current released mechanical data available for the designated device(s).

## 6.1 Package Thermal Resistance Characteristics

Table 6-1provides the estimated thermal resistance characteristics for the recommended package types used on the device.

**Table 6-1. Thermal Resistance Characteristics**

| PARAMETER | GGU PACKAGE | PGE PACKAGE | UNIT |
|---|---|---|---|
| $R_{\theta JA}$ | 38 | 56 | °C/W |
| $R_{\theta JC}$ | 5 | 5 | °C/W |

## PACKAGING INFORMATION

| Orderable Device | Status (1) | Package Type | Package Drawing | Pins | Package Qty | Eco Plan (2) | Lead/Ball Finish | MSL Peak Temp (3) |
|---|---|---|---|---|---|---|---|---|
| TMS320VC5416GGU120 | ACTIVE | BGA | GGU | 144 | 160 | TBD | SNPB | Level-3-220C-168HR |
| TMS320VC5416GGU160 | ACTIVE | BGA | GGU | 144 | 160 | TBD | SNPB | Level-3-220C-168HR |
| TMS320VC5416PGE120 | ACTIVE | LQFP | PGE | 144 | 60 | Green (RoHS & no Sb/Br) | CU NIPDAU | Level-1-260C-UNLIM |
| TMS320VC5416PGE160 | ACTIVE | LQFP | PGE | 144 | 1 | Green (RoHS & no Sb/Br) | CU NIPDAU | Level-1-260C-UNLIM |
| TMS320VC5416ZGU120 | ACTIVE | BGA | ZGU | 144 | 160 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168HR |
| TMS320VC5416ZGU160 | ACTIVE | BGA | ZGU | 144 | 160 | Green (RoHS & no Sb/Br) | SNAGCU | Level-3-260C-168HR |
| TMSDVC5416GGUR160 | ACTIVE | BGA | GGU | 144 | 1000 | TBD | SNPB | Level-3-220C-168HR |

(1) The marketing status values are defined as follows:
**ACTIVE:** Product device recommended for new designs.
**LIFEBUY:** TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
**NRND:** Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
**PREVIEW:** Device has been announced but is not in production. Samples may or may not be available.
**OBSOLETE:** TI has discontinued the production of the device.

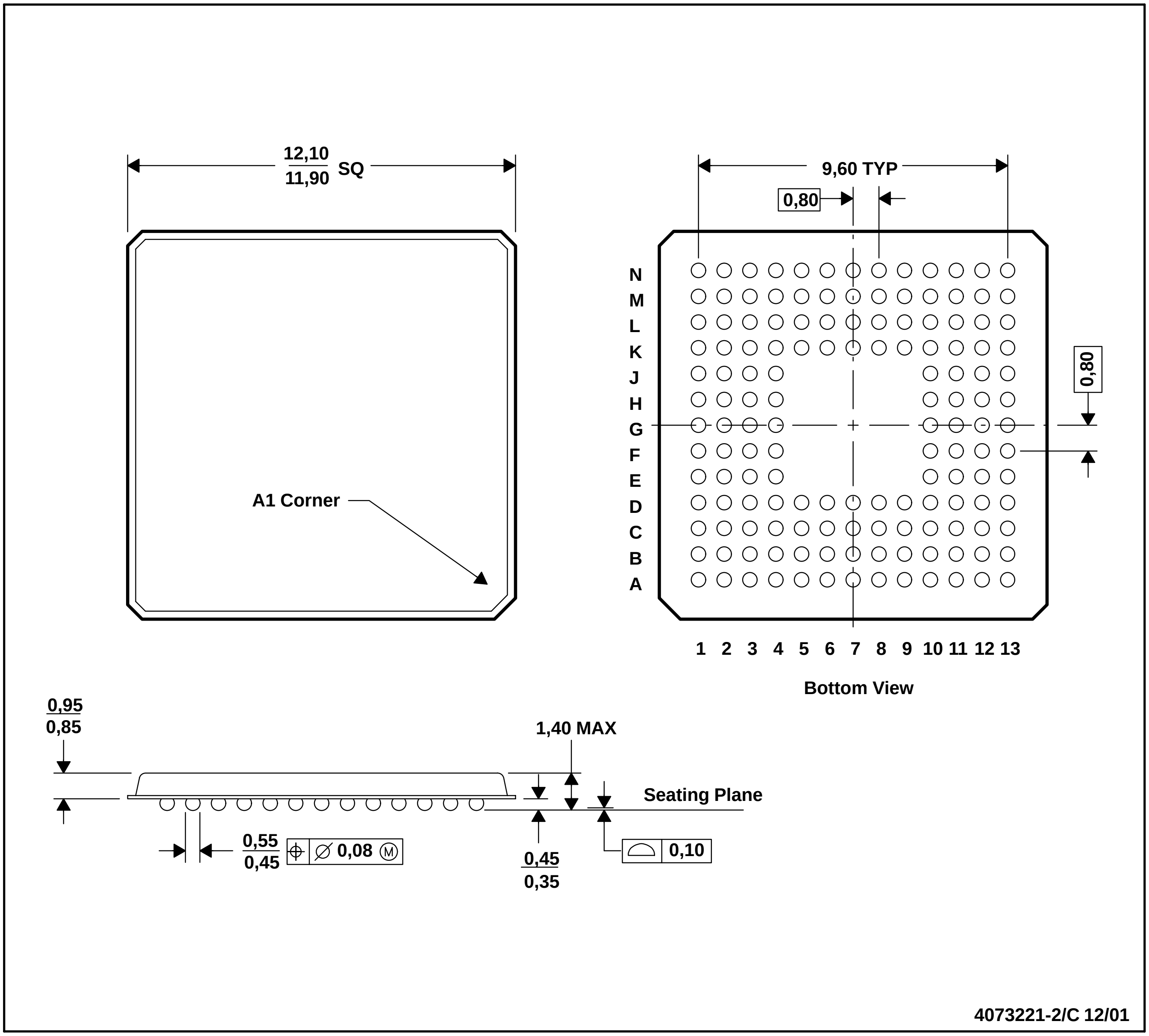
(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS) or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
**TBD:** The Pb-Free/Green conversion plan has not been defined.
**Pb-Free (RoHS):** TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
**Green (RoHS & no Sb/Br):** TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

(3) MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

**Important Information and Disclaimer:**The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

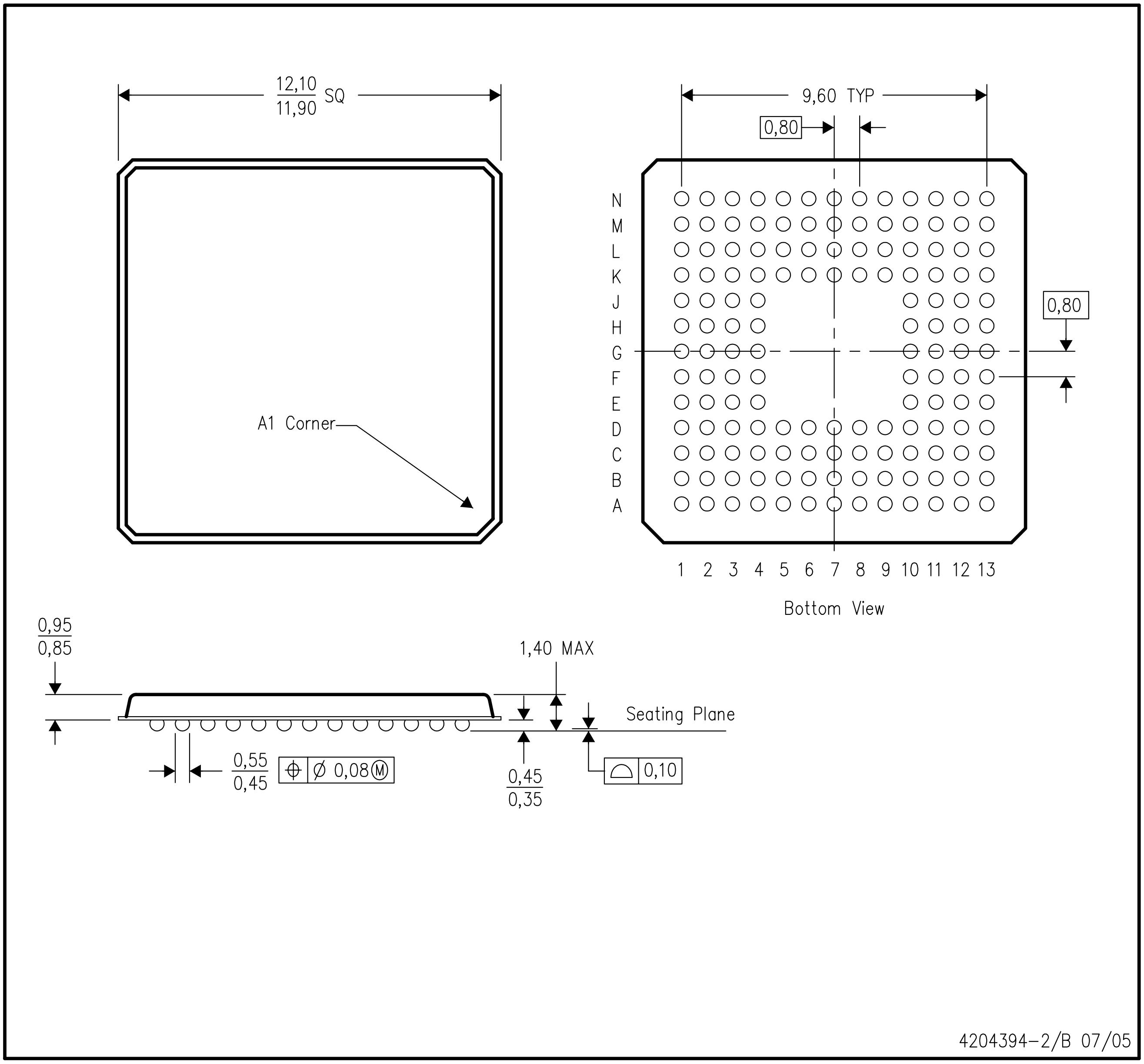
**GGU (S–PBGA–N144)** **PLASTIC BALL GRID ARRAY**




NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice
C. MicroStar BGA™ configuration

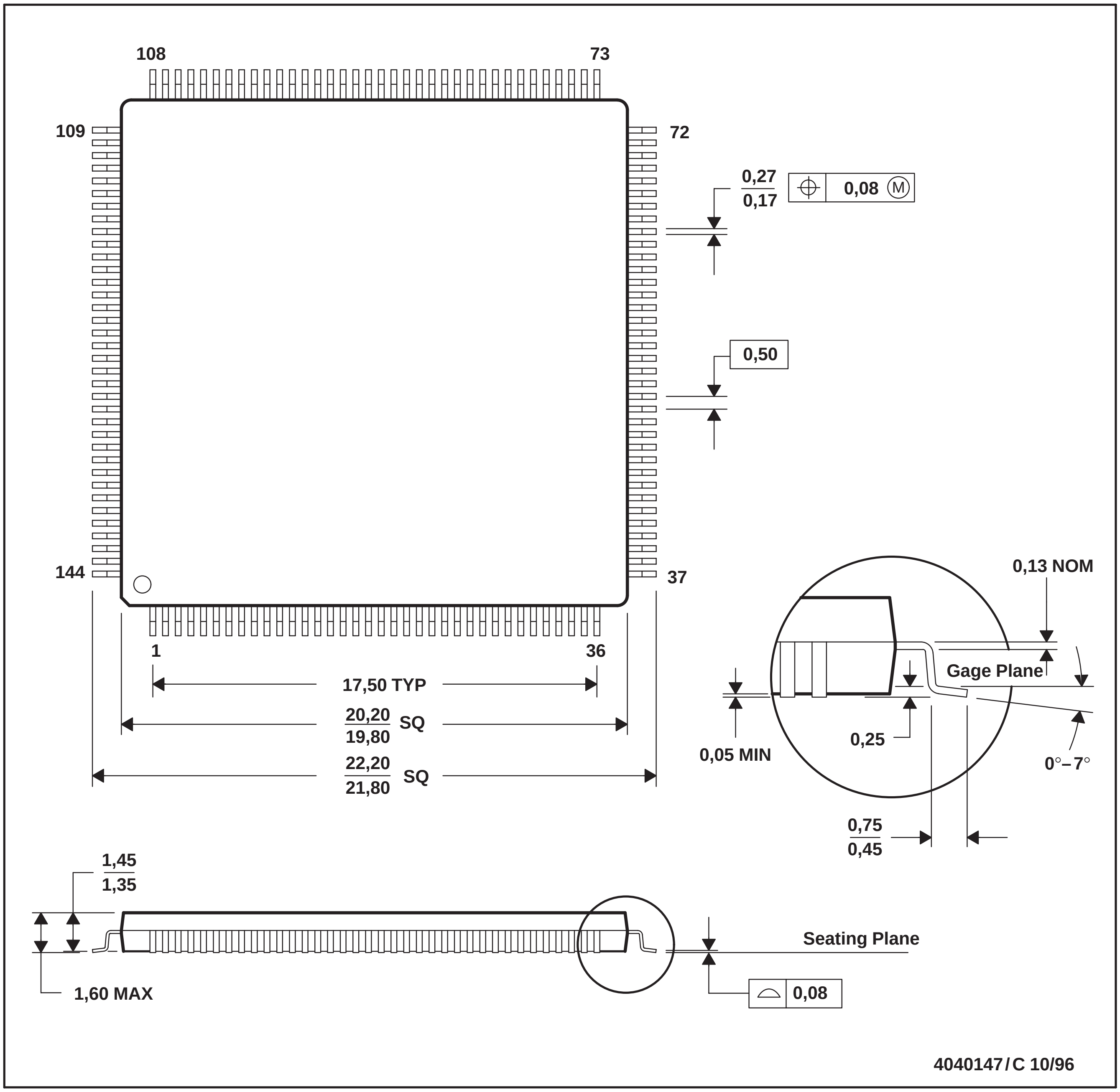
MicroStar BGA is a trademark of Texas Instruments Incorporated.

ZGU (S–PBGA–N144) PLASTIC BALL GRID ARRAY




NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Micro Star BGA configuration
D. This is a lead–free solder ball design.

**PGE (S-PQFP-G144)** **PLASTIC QUAD FLATPACK**




NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-026

## IMPORTANT NOTICE



Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

| **Products** | | **Applications** | |
|---|---|---|---|
| Amplifiers | amplifier.ti.com | Audio | www.ti.com/audio |
| Data Converters | dataconverter.ti.com | Automotive | www.ti.com/automotive |
| DSP | dsp.ti.com | Broadband | www.ti.com/broadband |
| Interface | interface.ti.com | Digital Control | www.ti.com/digitalcontrol |
| Logic | logic.ti.com | Military | www.ti.com/military |
| Power Mgmt | power.ti.com | Optical Networking | www.ti.com/opticalnetwork |
| Microcontrollers | microcontroller.ti.com | Security | www.ti.com/security |
| | | Telephony | www.ti.com/telephony |
| | | Video & Imaging | www.ti.com/video |
| | | Wireless | www.ti.com/wireless |

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265